BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

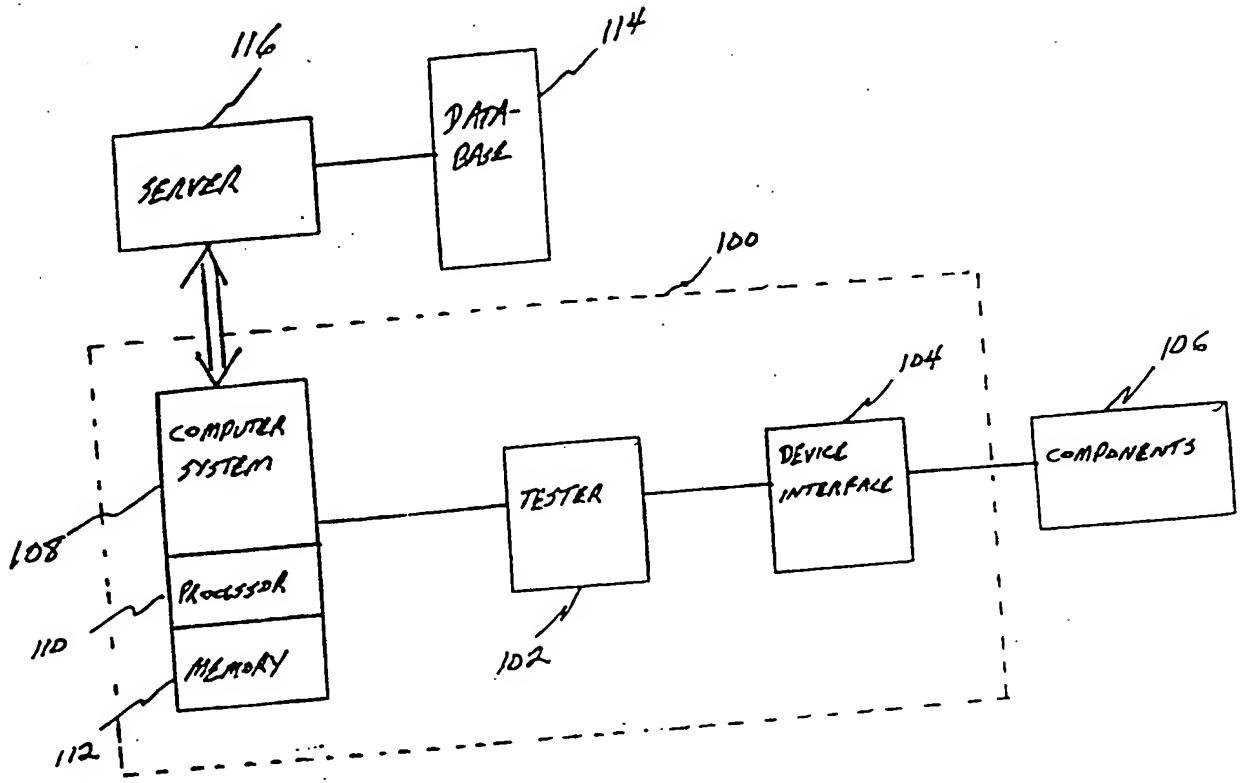

116
SERVER
114
DATA-BASE
100
COMPUTER SYSTEM
108
PROCESSOR
110
MEMORY
112
TESTER
102
104
DEVICE INTERFACE
106
COMPONENTS

FIG. 1

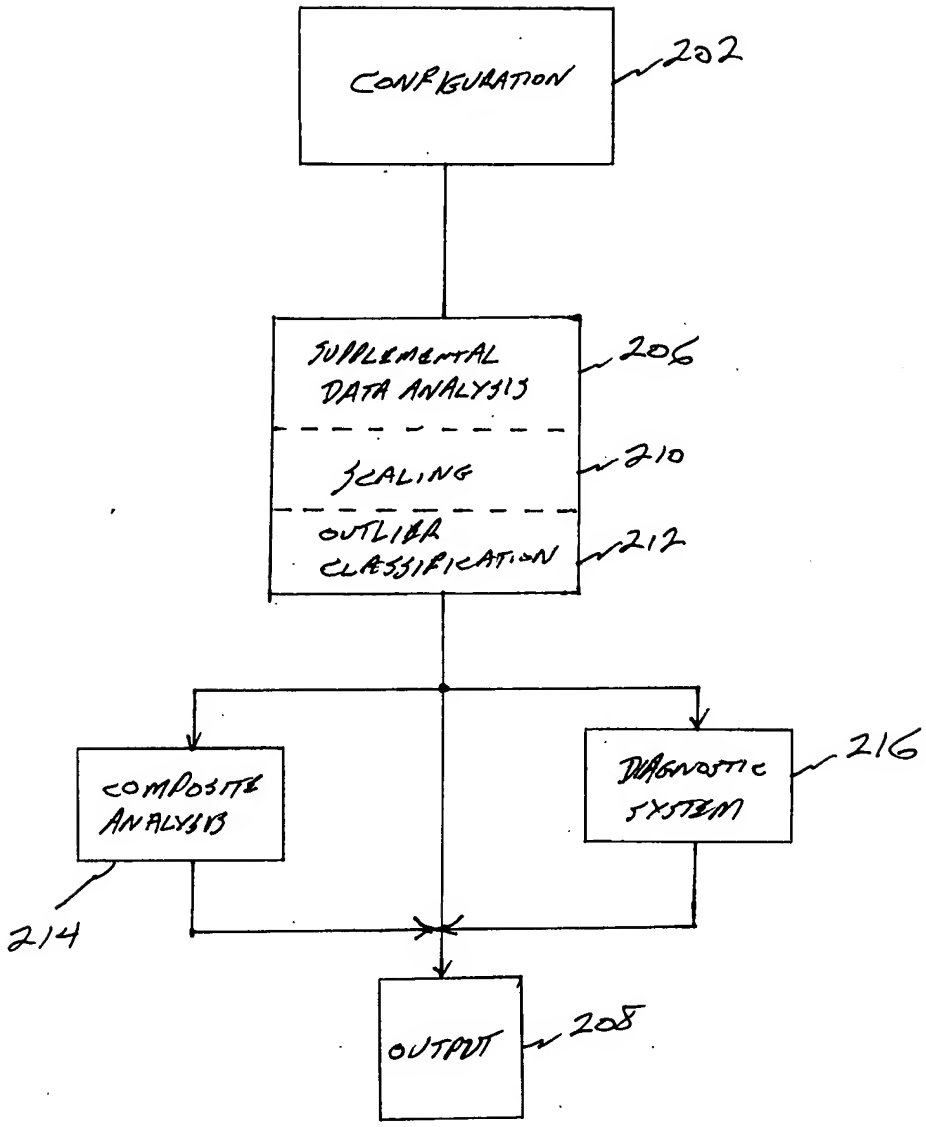

CONFIGURATION
202
SUPPLEMENTAL DATA ANALYSIS
206
SCALING
210
OUTLIER CLASSIFICATION
212
COMPOSITE ANALYSIS
214
DIAGNOSTIC SYSTEM
216
OUTPUT
208

FIG. 2

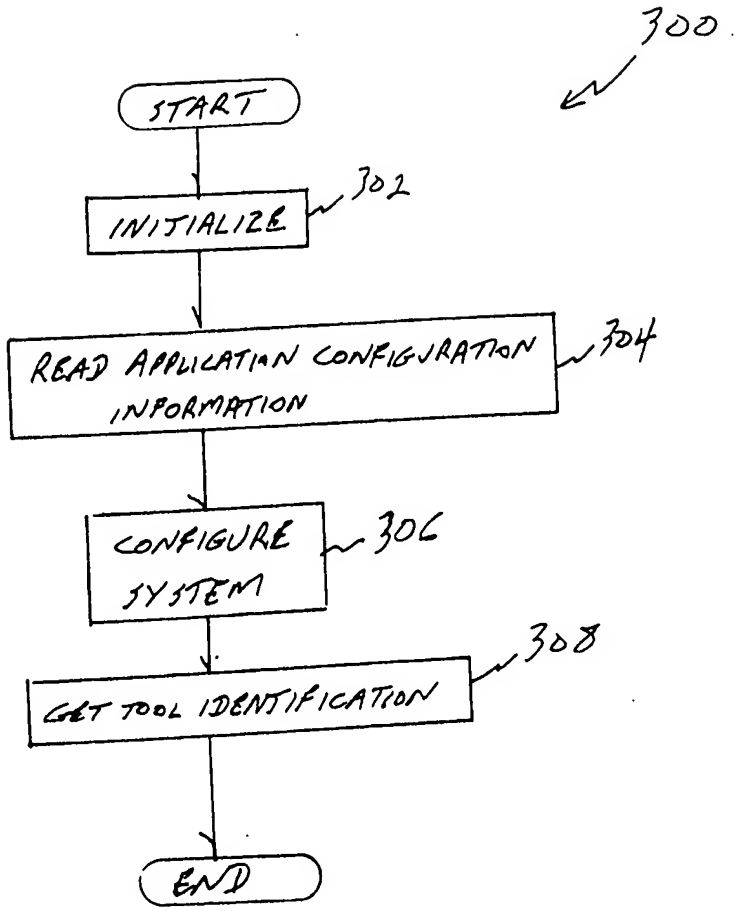

300
START
INITIALIZE
302
READ APPLICATION CONFIGURATION INFORMATION
304
CONFIGURE SYSTEM
306
GET TOOL IDENTIFICATION
308
END

FIG. 3

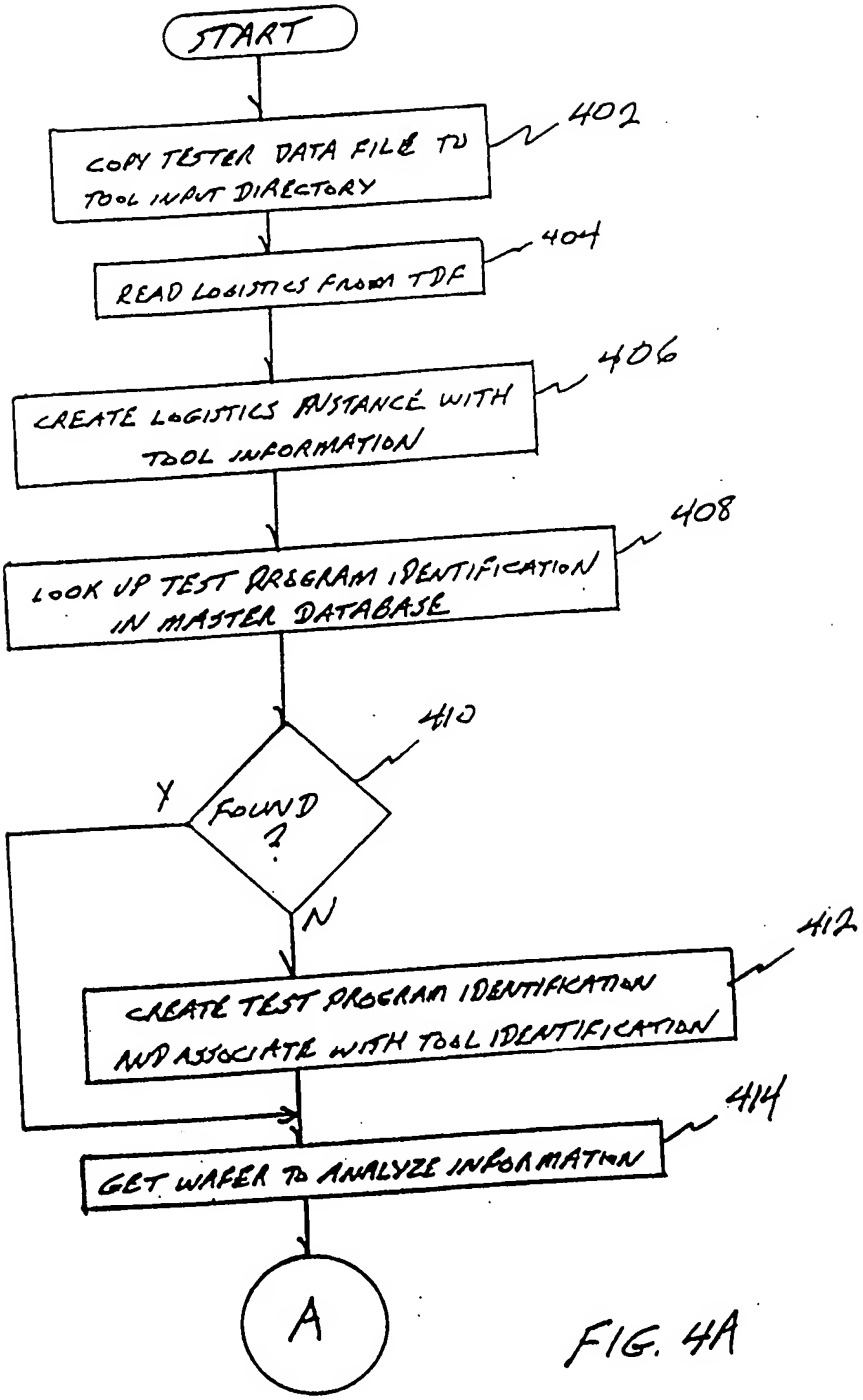

START
COPY TESTER DATA FILE TO TOOL INPUT DIRECTORY
402
READ LOGISTICS FROM TDF
404
CREATE LOGISTICS INSTANCE WITH TOOL INFORMATION
406
LOOK UP TEST PROGRAM IDENTIFICATION IN MASTER DATABASE
408
FOUND ?
410
Y
N
CREATE TEST PROGRAM IDENTIFICATION AND ASSOCIATE WITH TOOL IDENTIFICATION
412
GET WAFER TO ANALYZE INFORMATION
414
A

FIG. 4A

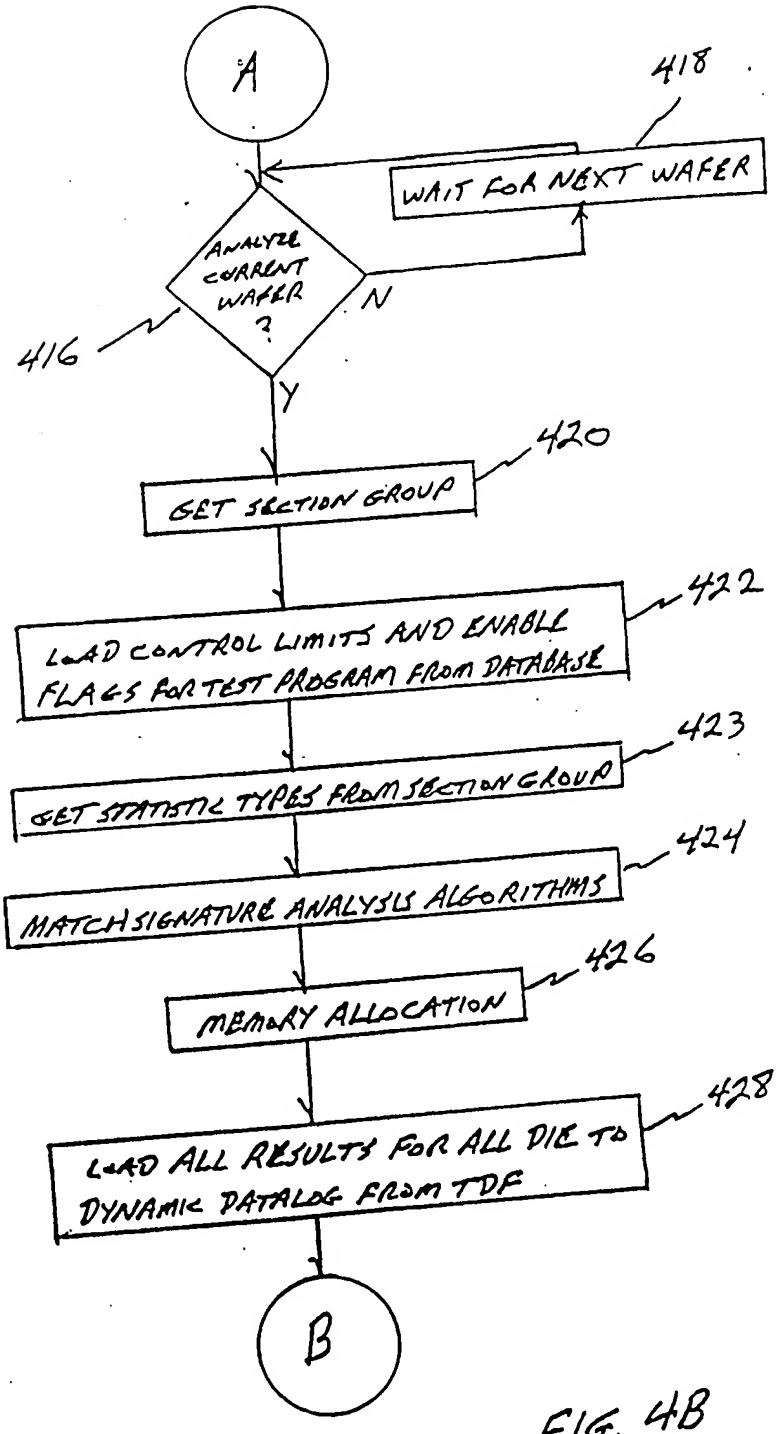

A
418
WAIT FOR NEXT WAFER
ANALYZE CURRENT WAFER ?
N
416
Y
420
GET SECTION GROUP
422
LOAD CONTROL LIMITS AND ENABLE FLAGS FOR TEST PROGRAM FROM DATABASE
423
GET STATISTIC TYPES FROM SECTION GROUP
424
MATCH SIGNATURE ANALYSIS ALGORITHMS
426
MEMORY ALLOCATION
428
LOAD ALL RESULTS FOR ALL DIE TO DYNAMIC DATALOG FROM TDF
B

FIG. 4B

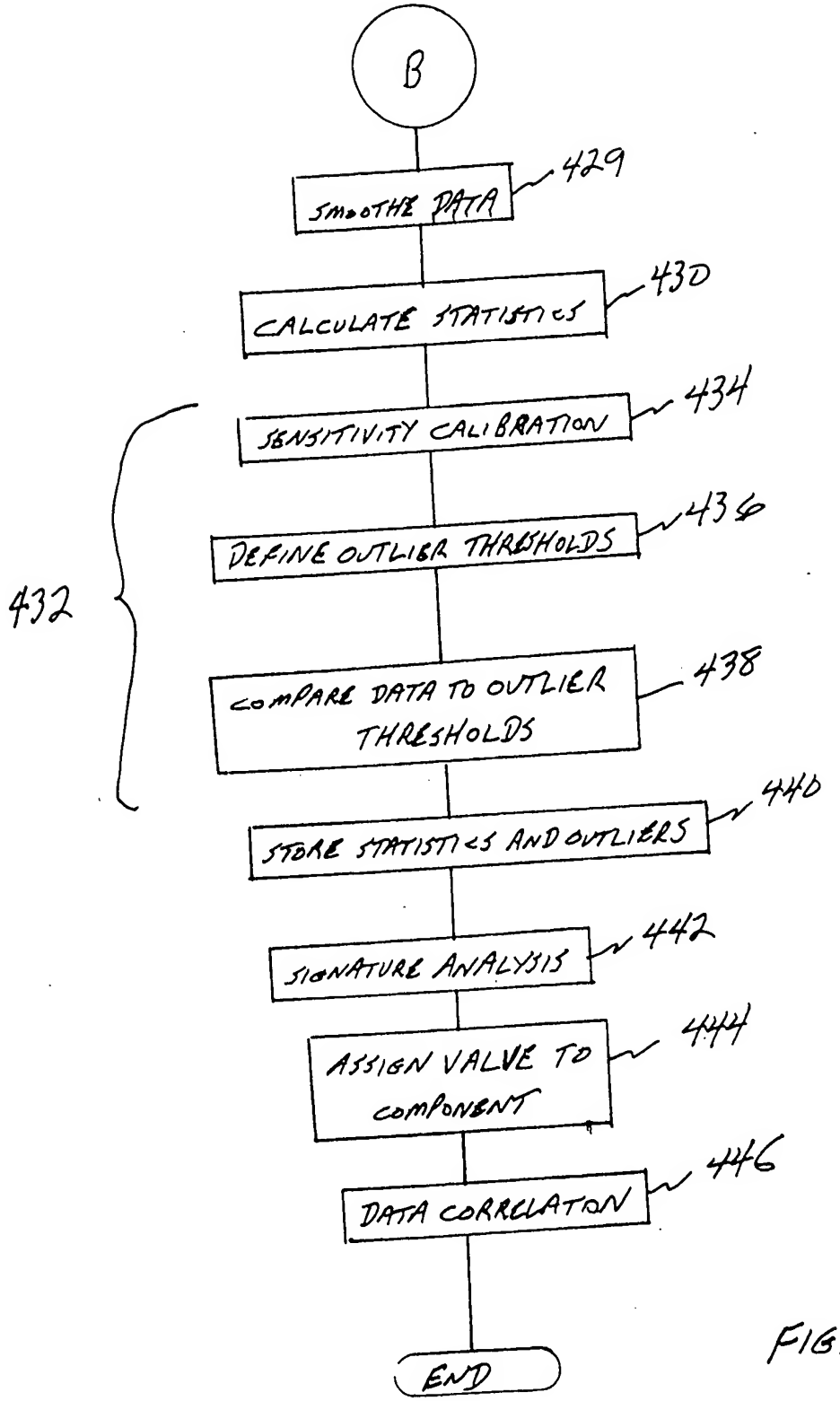

B
SMOOTHE DATA
429
CALCULATE STATISTICS
430
SENSITIVITY CALIBRATION
434
DEFINE OUTLIER THRESHOLDS
436
432
COMPARE DATA TO OUTLIER THRESHOLDS
438
STORE STATISTICS AND OUTLIERS
440
SIGNATURE ANALYSIS
442
ASSIGN VALVE TO COMPONENT
444
DATA CORRELATION
446
END

FIG. 4C

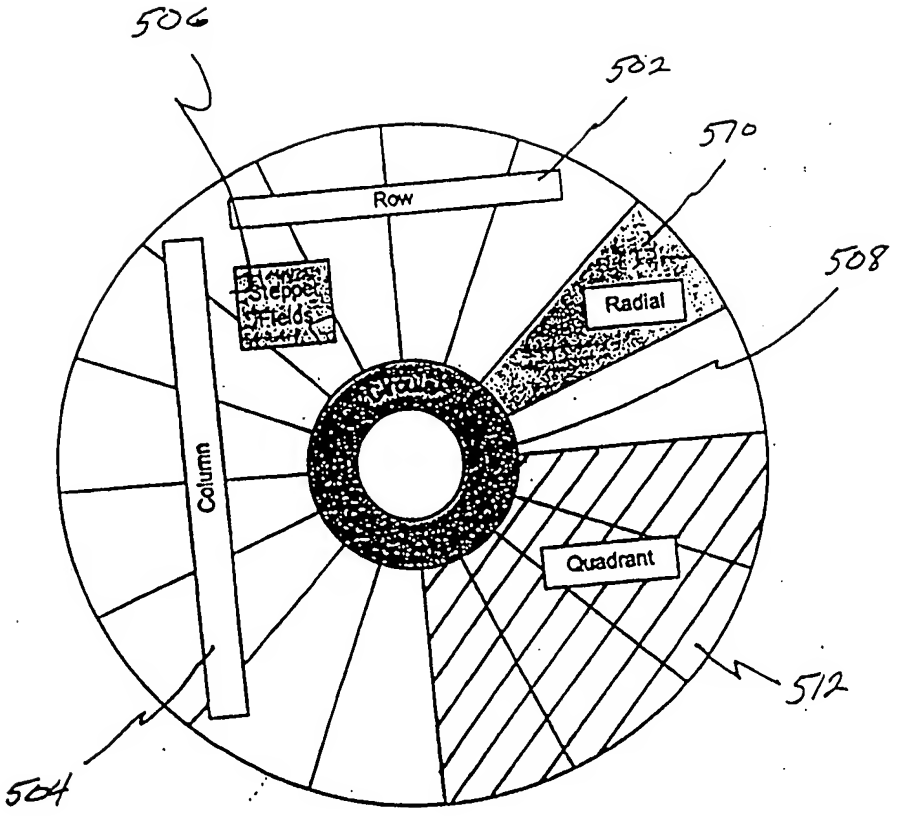

506
502
570
508
Row
Stepper Fields
Radial
Pupil
Column
Quadrant
512
504

FIG. 5

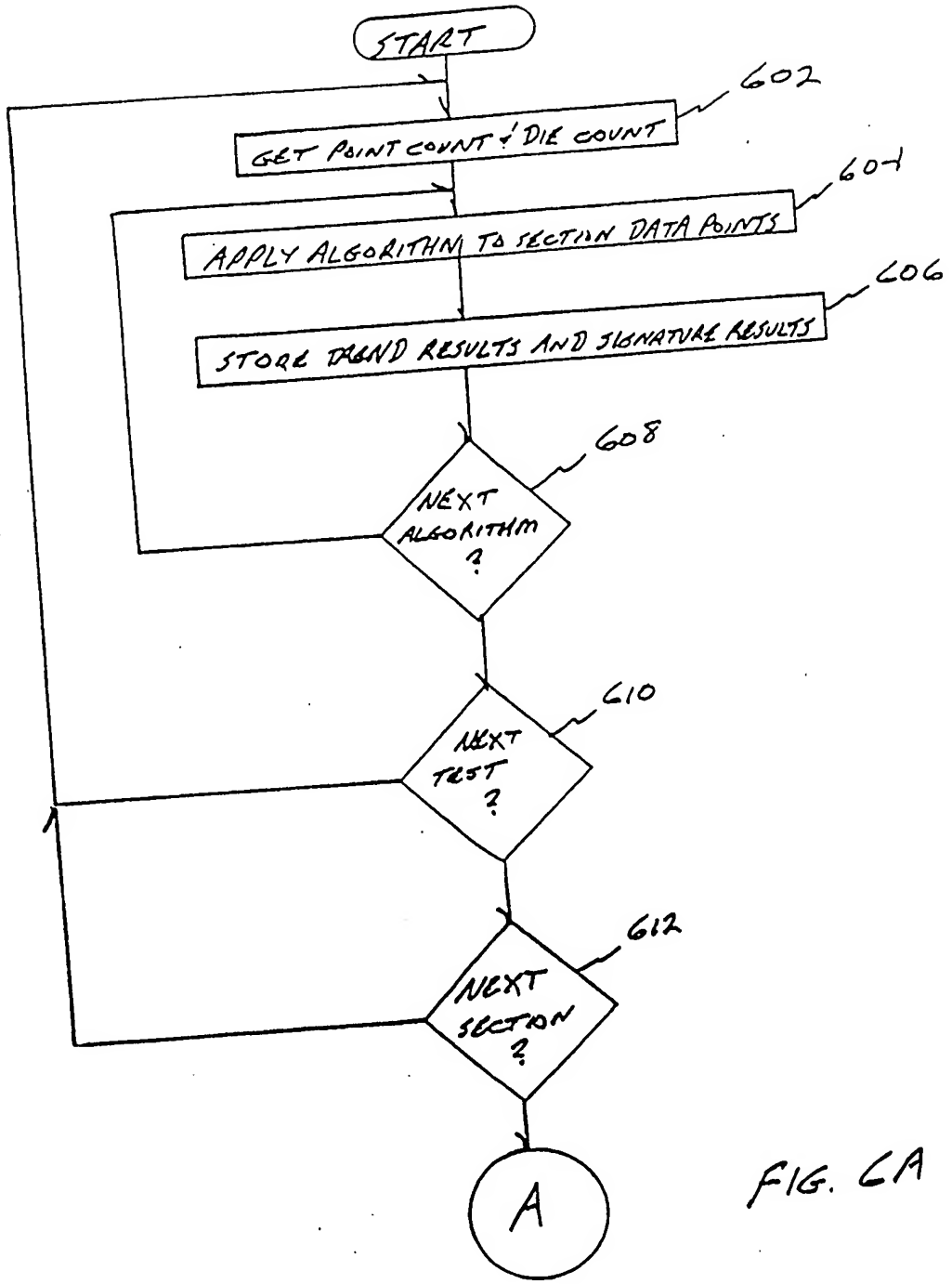

START
602
GET POINT COUNT & DIE COUNT
604
APPLY ALGORITHM TO SECTION DATA POINTS
606
STORE DIGND RESULTS AND SIGNATURE RESULTS
608
NEXT ALGORITHM ?
610
NEXT TEST ?
612
NEXT SECTION ?
A

FIG. 6A

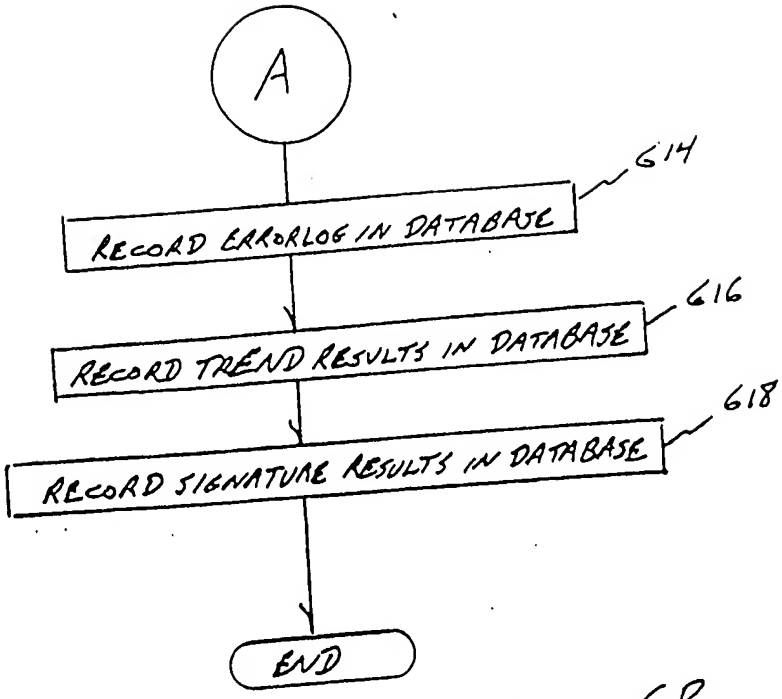

A
614
RECORD ERRORLOG IN DATABASE
616
RECORD TREND RESULTS IN DATABASE
618
RECORD SIGNATURE RESULTS IN DATABASE
END

FIG. 6B

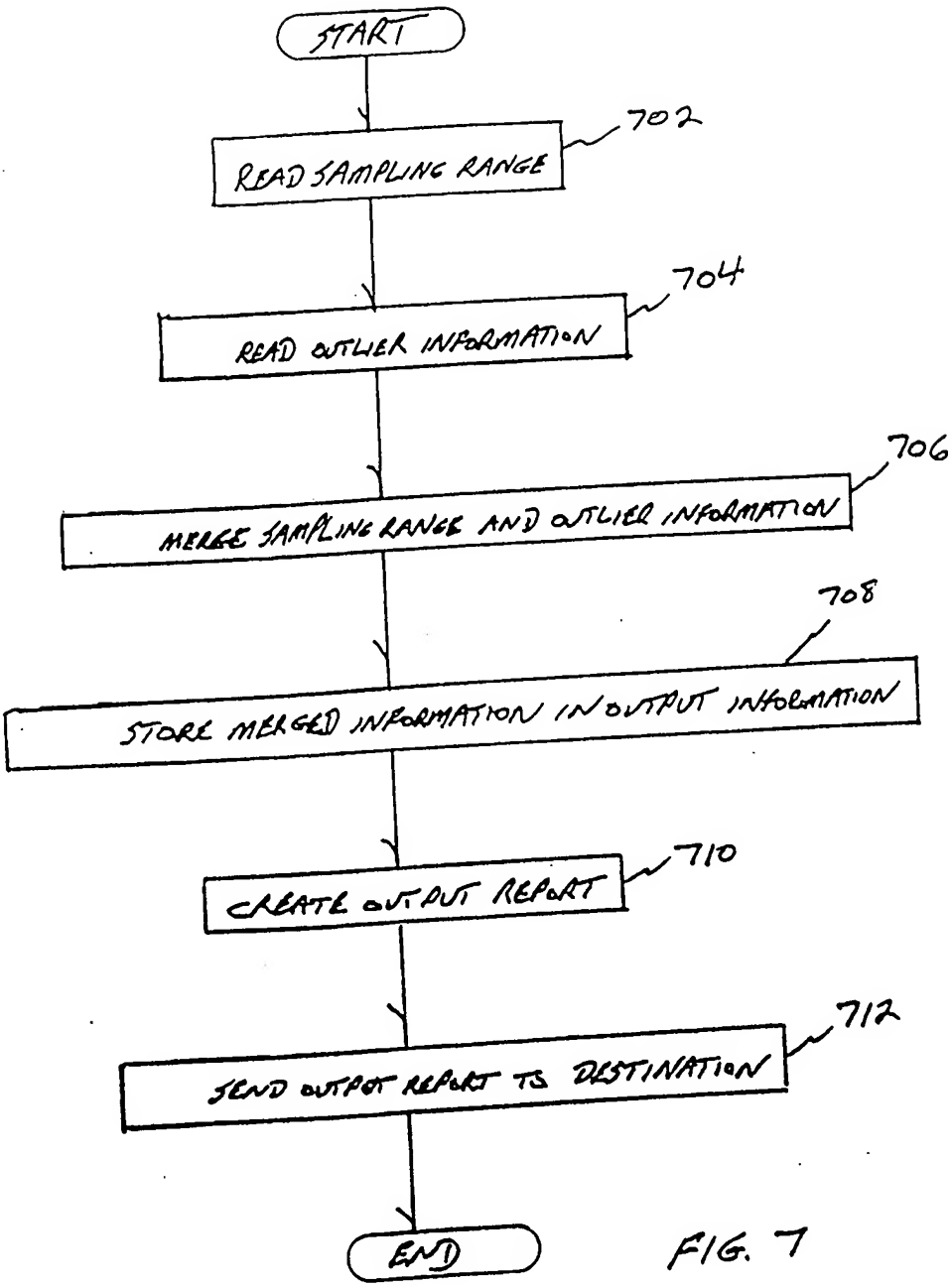

START
READ SAMPLING RANGE
702
READ OUTLIER INFORMATION
704
MERGE SAMPLING RANGE AND OUTLIER INFORMATION
706
STORE MERGED INFORMATION IN OUTPUT INFORMATION
708
CREATE OUTPUT REPORT
710
SEND OUTPUT REPORT TO DESTINATION
712
END

FIG. 7

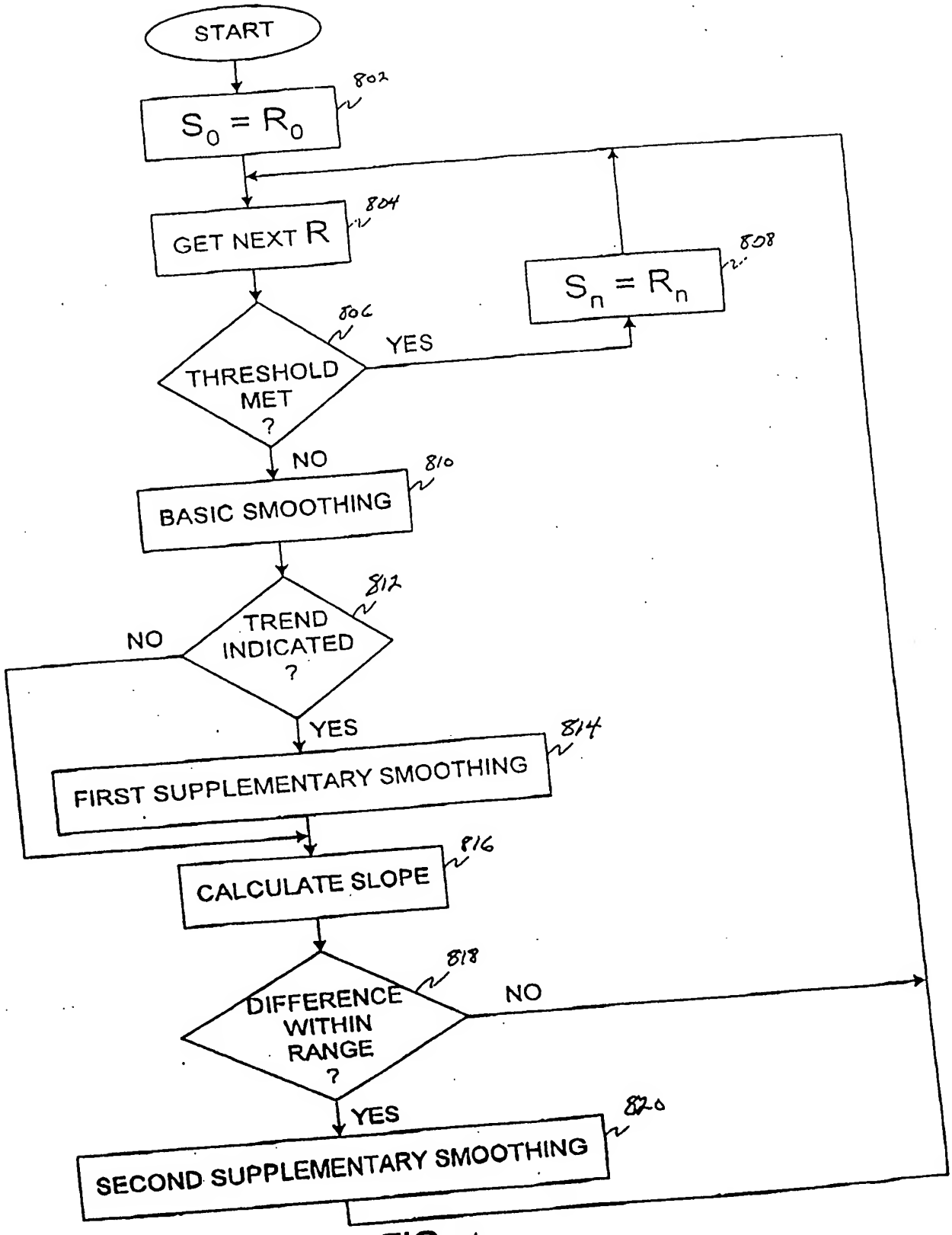
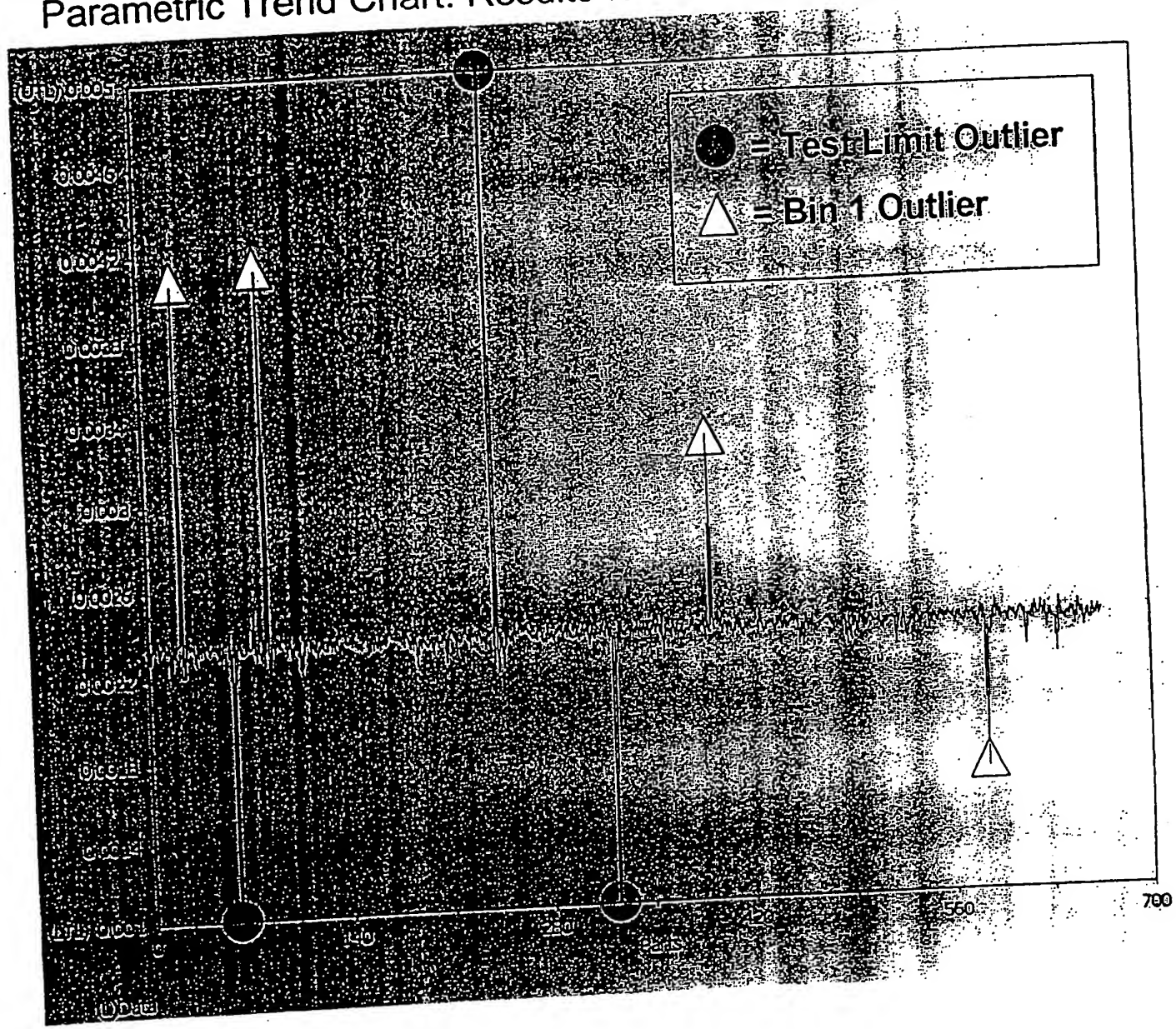

START
802
$S_0 = R_0$
804
GET NEXT R
806
THRESHOLD MET ?
YES
808
$S_n = R_n$
NO
810
BASIC SMOOTHING
812
TREND INDICATED ?
NO
YES
814
FIRST SUPPLEMENTARY SMOOTHING
816
CALCULATE SLOPE
818
DIFFERENCE WITHIN RANGE ?
NO
YES
820
SECOND SUPPLEMENTARY SMOOTHING

FIG. 8

FIG. 9


Parametric Trend Chart: Results for one test, for an entire wafer
= Test Limit Outlier
= Bin 1 Outlier
560
700

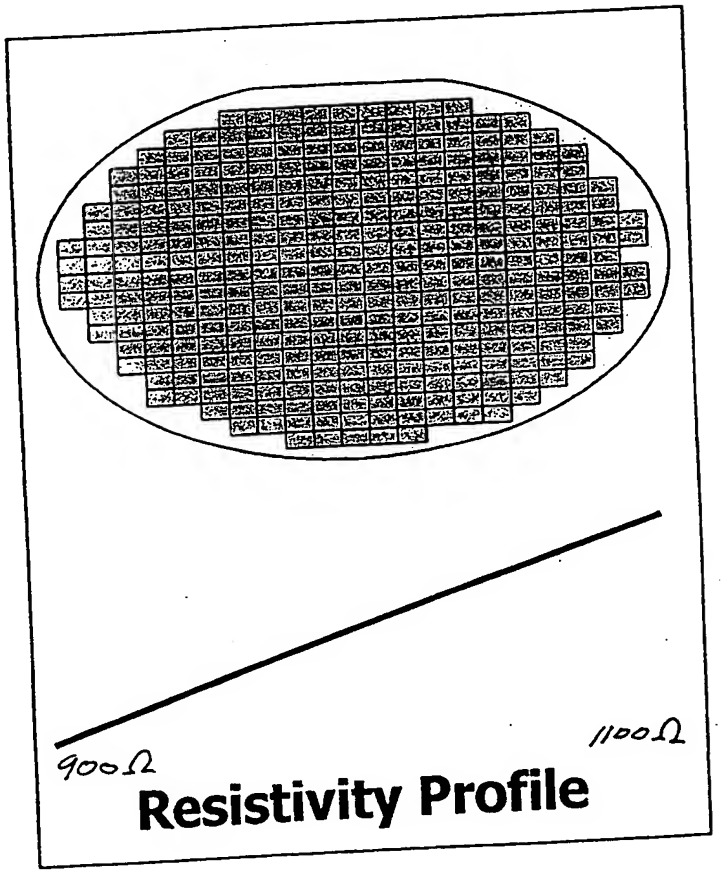


FIG. 10

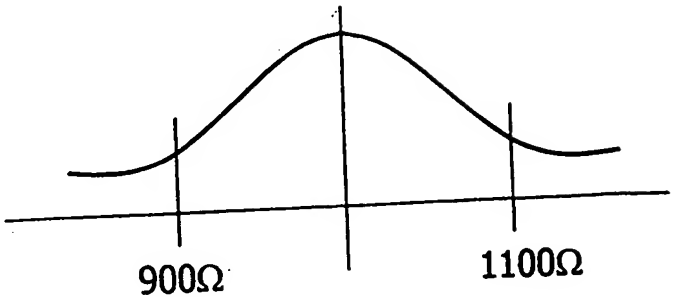


FIG. 11

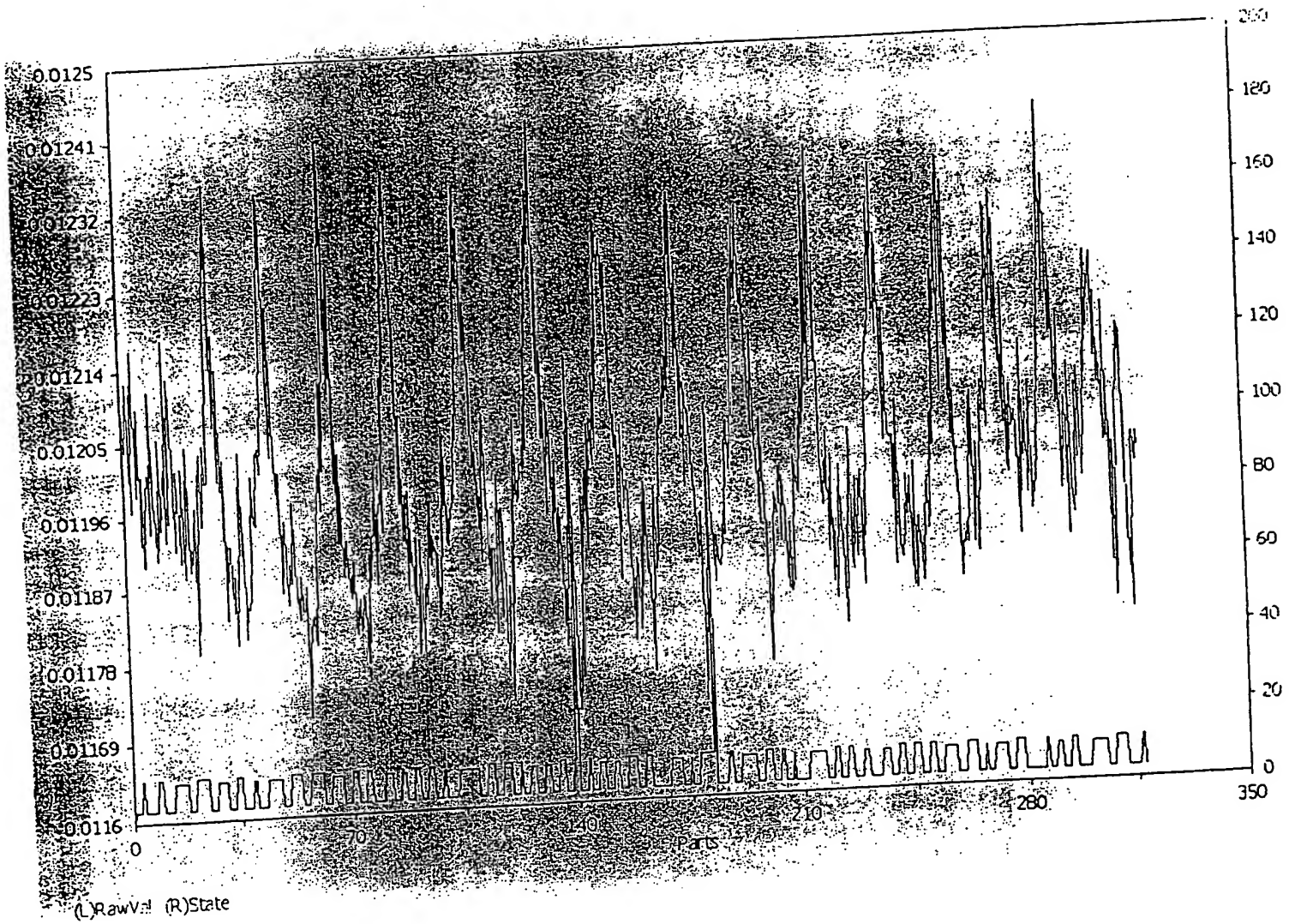

0.0125
0.01241
0.01232
0.01223
0.01214
0.01205
0.01196
0.01187
0.01178
0.01169
0.0116
0
70
140
210
280
350
Pts
(L)RawV (R)State

FIG. 12A

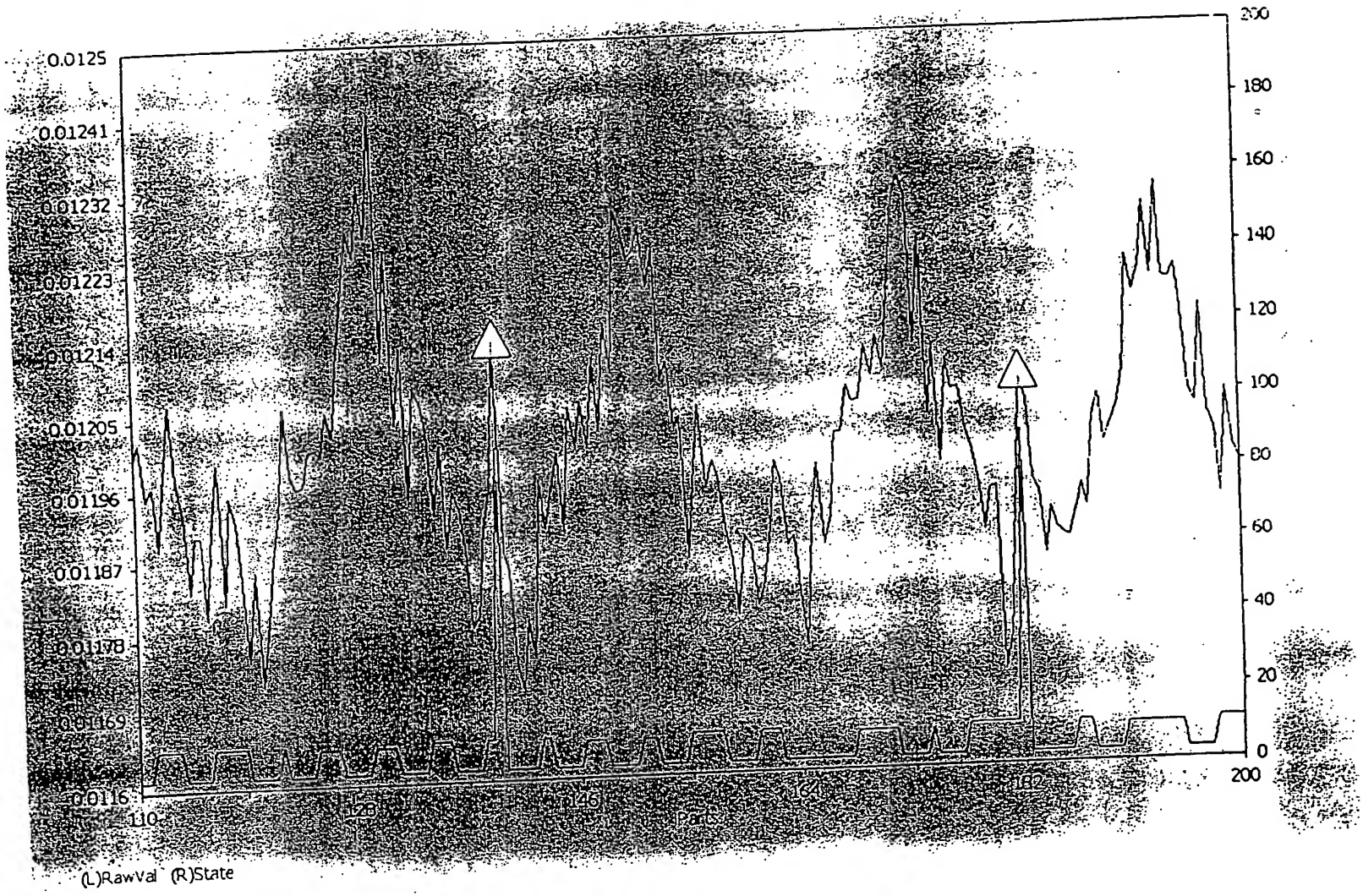

0.0125
0.01241
0.01232
0.01223
0.01214
0.01205
0.01196
0.01187
0.01178
0.01169
0.0116
110
200
180
160
140
120
100
80
60
40
20
0
200
(L)RawVal (R)State

FIG. 12B

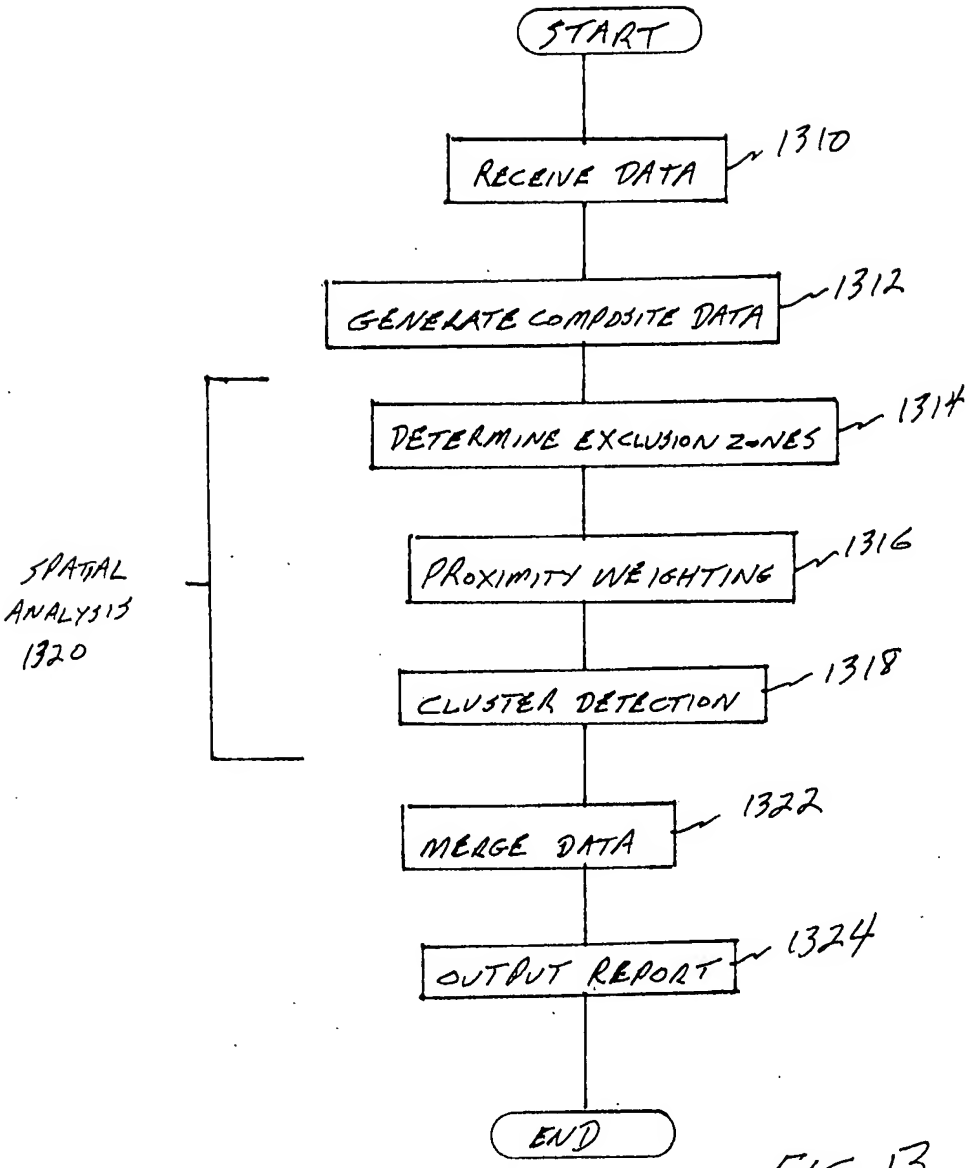

START
RECEIVE DATA
1310
GENERATE COMPOSITE DATA
1312
DETERMINE EXCLUSION ZONES
1314
PROXIMITY WEIGHTING
1316
CLUSTER DETECTION
1318
SPATIAL ANALYSIS 1320
MERGE DATA
1322
OUTPUT REPORT
1324
END

FIG. 13

Composite Addition

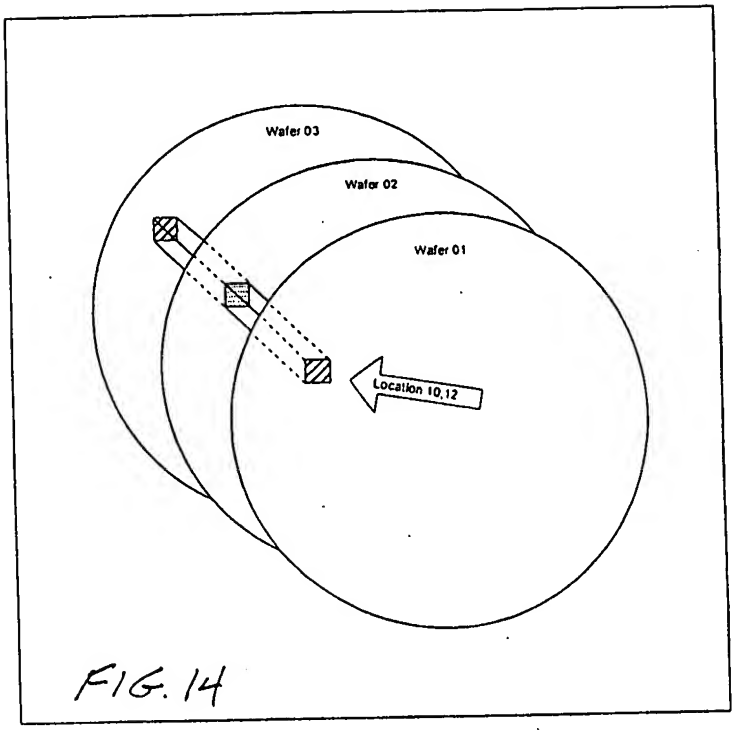


FIG. 14

Legend

Devices with value of 5

Devices with value of 10

Example:

For the location 10,12 a variable with the definition of five would have a value of two and a variable with a definition of ten wold have a value of one.

Now, assume that the user has a fomula of X > 1 and also assume that x is a variable with a definition value of five According ot the data, x would have a value of 2 (as we saw above) and the formula would equate as true. Since the formula resulted in a true value, it would be included in the composite map.

Cumulative Squared Method

Done on X,Y location basis for entire dataset / Lot

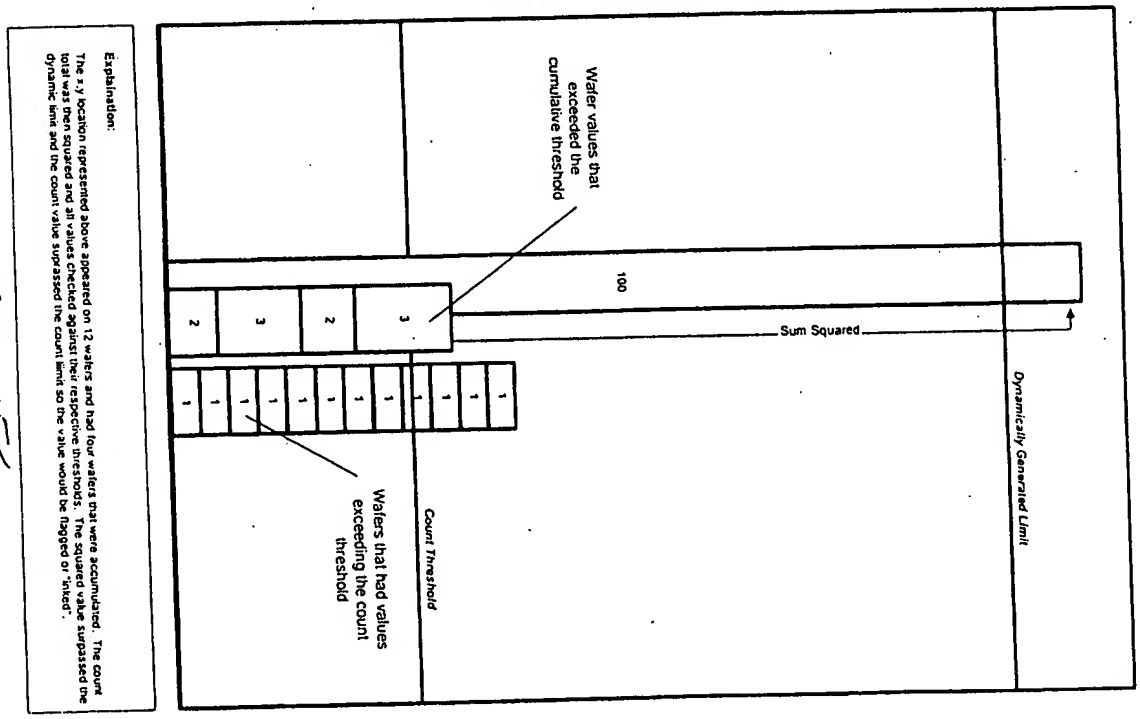


FIG. 15

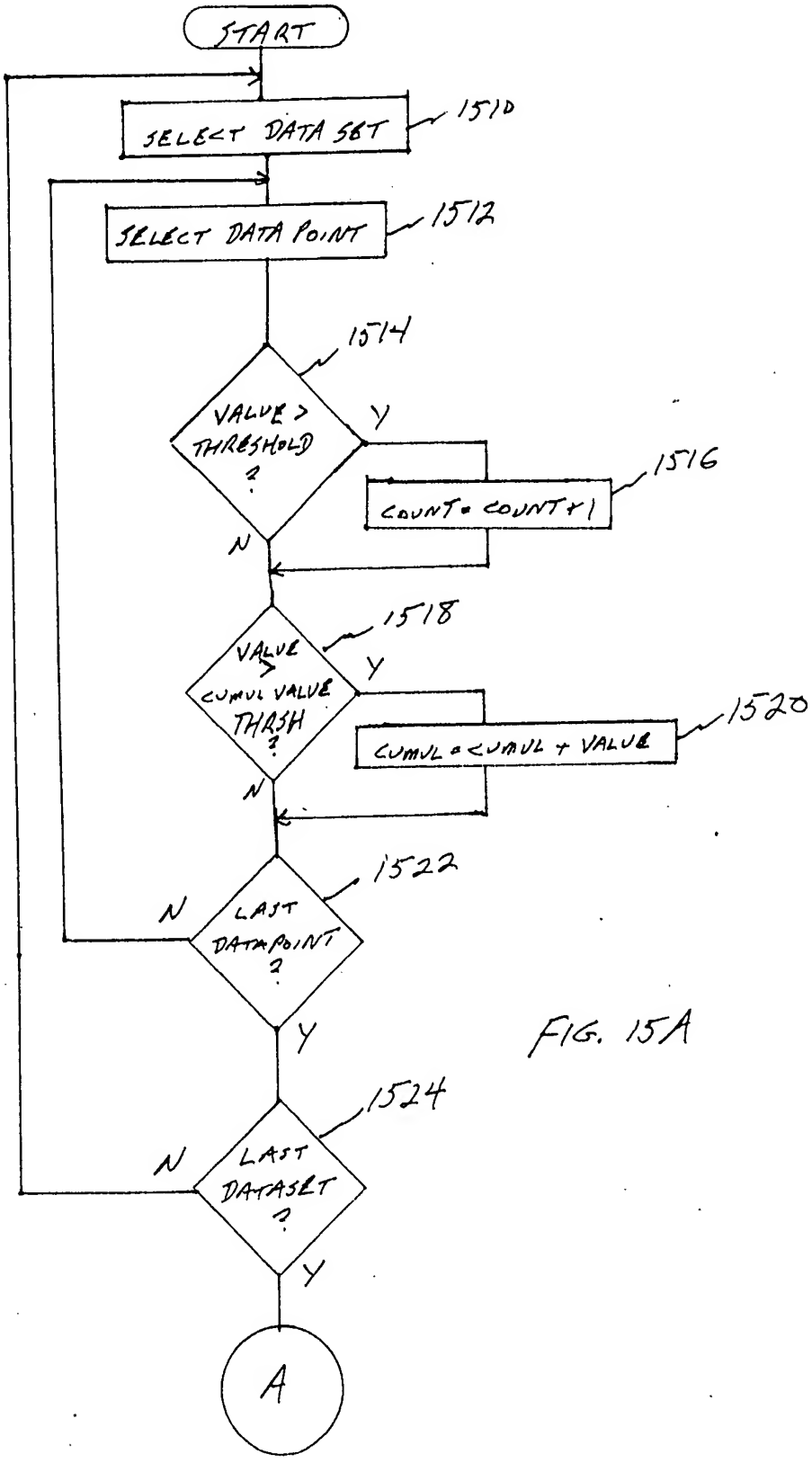

START
SELECT DATA SET
1510
SELECT DATA POINT
1512
1514
VALUE > THRESHOLD ?
Y
N
COUNT = COUNT + 1
1516
1518
VALUE > CUMUL VALUE THRSH ?
Y
N
CUMUL = CUMUL + VALUE
1520
1522
LAST DATA POINT ?
N
Y
1524
LAST DATASET ?
N
Y
A

FIG. 15A

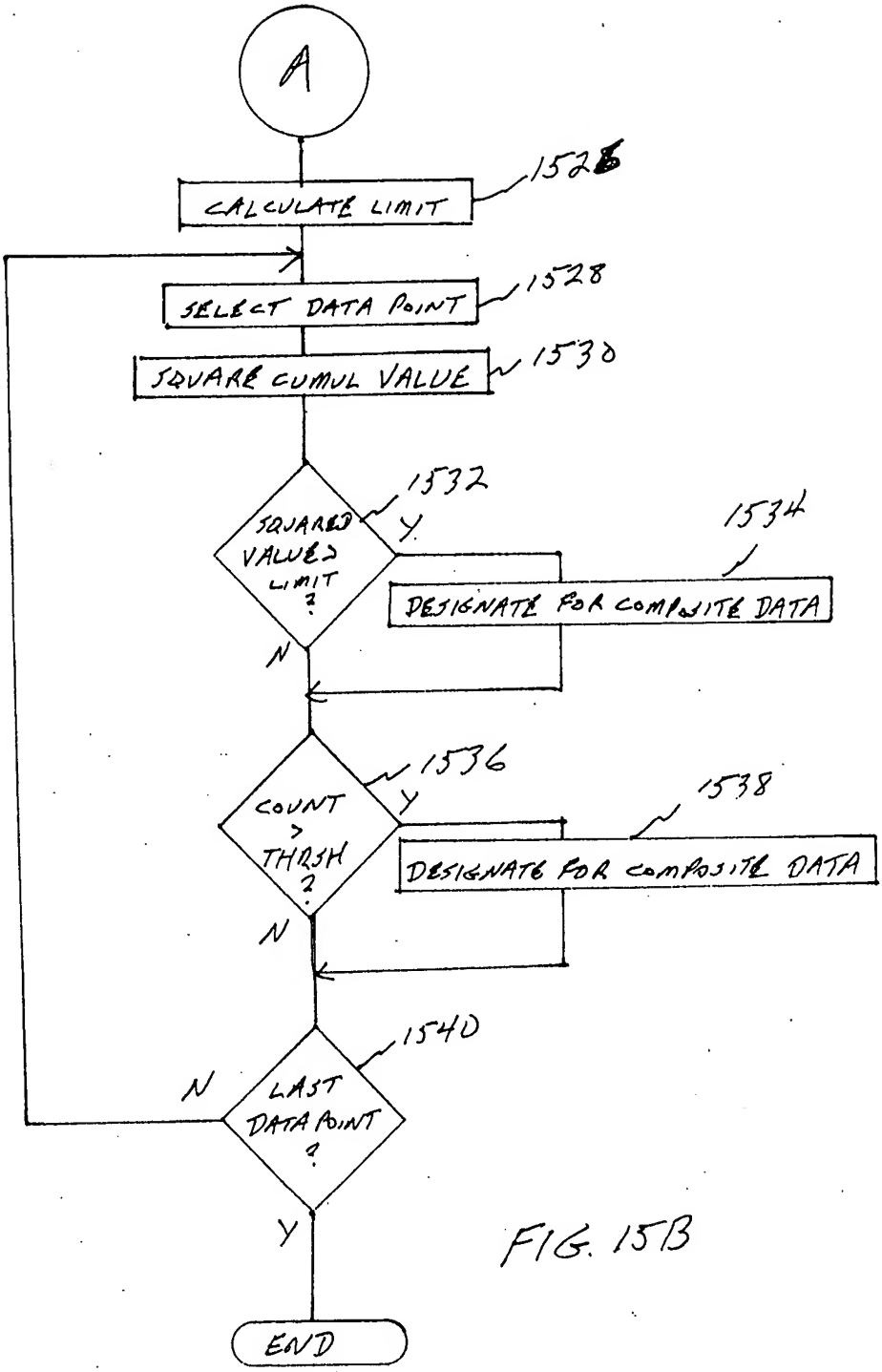
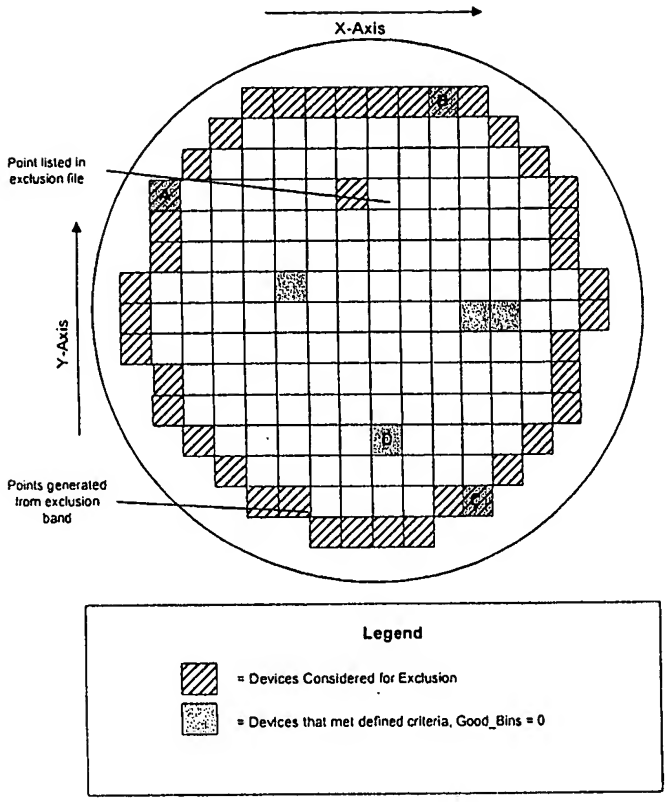

A
CALCULATE LIMIT
1526
SELECT DATA POINT
1528
SQUARE CUMUL VALUE
1530
SQUARED VALUE > LIMIT ?
1532
Y
DESIGNATE FOR COMPOSITE DATA
1534
N
COUNT > THRSH ?
1536
Y
DESIGNATE FOR COMPOSITE DATA
1538
N
LAST DATA POINT ?
1540
N
Y
END

FIG. 15B

Exclusion Zone Graphical Representation



Description

The exclusion zone function is designed exclude certain devices based on specific x,y locations read from a file or devices within a certain distance of the edge of the dataset that meet a specified criteria. Devices are excluded by setting them to a value which is out of the range of subsequent processes such as proximity weighting.

Logic

Loop through each x,y location
If x,y location is within EXCLUSION_RANGE of the edge of the data set then
Check is x,y location value (bin) meets EXCLUSION_CRITERIA
If EXCLUSION_CRITERIA is met, set x,y location to EXCLUSION_INK_VALUE

Process EXCLUSION_FILE and set all x,y locations listed in file to EXCLUSION_INK_VALUE

Examples

Device A would be excluded
Device B would be excluded
Device C would be excluded
Device D would be included

Parameters

EXCLUSION_RANGE = 1
EXCLUSION_INK_VALUE = -1
EXCLUSION_CRITERIA = "Good_Bins = 0"
EXCLUSION_FILE = "D:/Step2/my.exclude"

FIG. 16

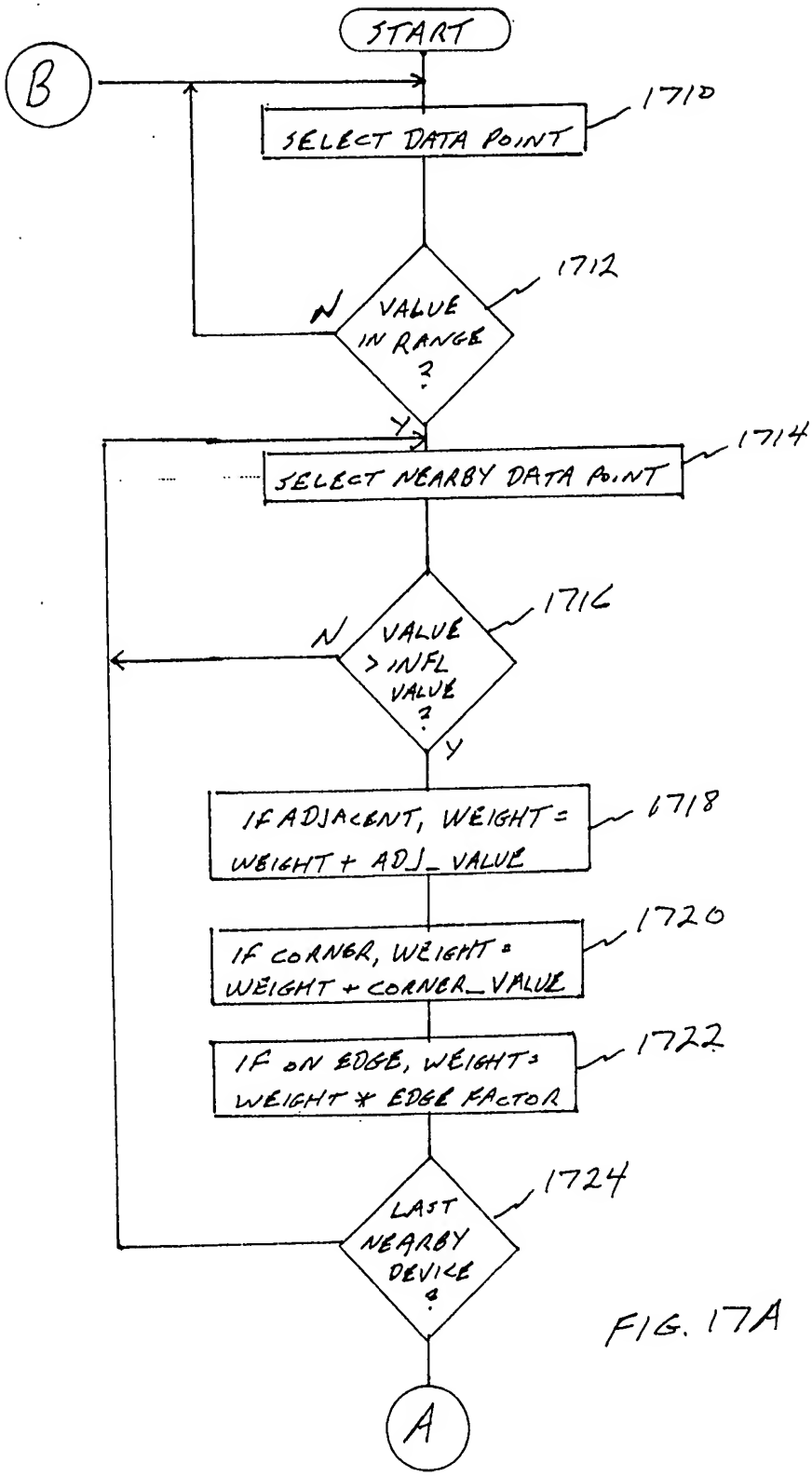

START
B
SELECT DATA POINT
1710
VALUE IN RANGE ?
1712
N
Y
SELECT NEARBY DATA POINT
1714
VALUE > INFL VALUE ?
1716
N
Y
IF ADJACENT, WEIGHT = WEIGHT + ADJ_VALUE
1718
IF CORNER, WEIGHT = WEIGHT + CORNER_VALUE
1720
IF ON EDGE, WEIGHT = WEIGHT * EDGE FACTOR
1722
LAST NEARBY DEVICE ?
1724
A

FIG. 17A

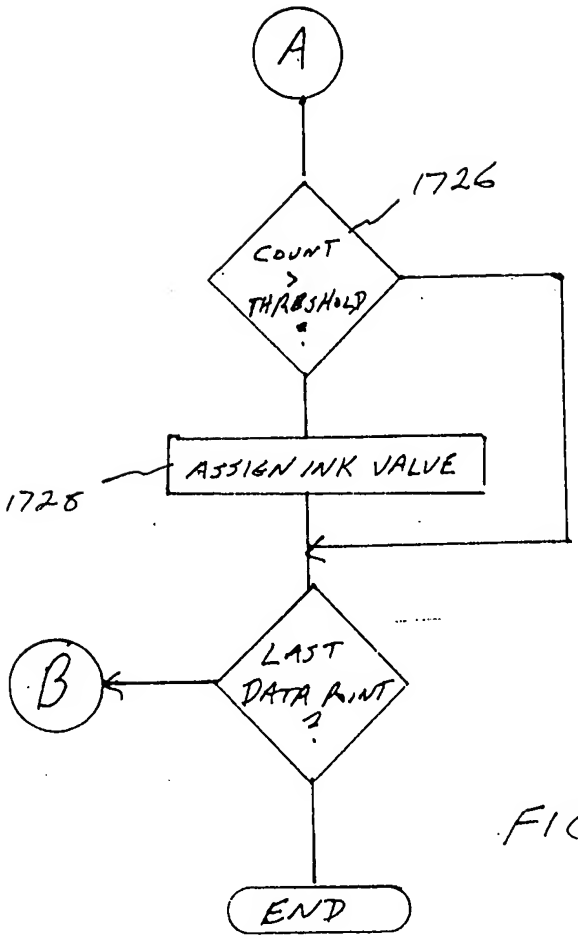
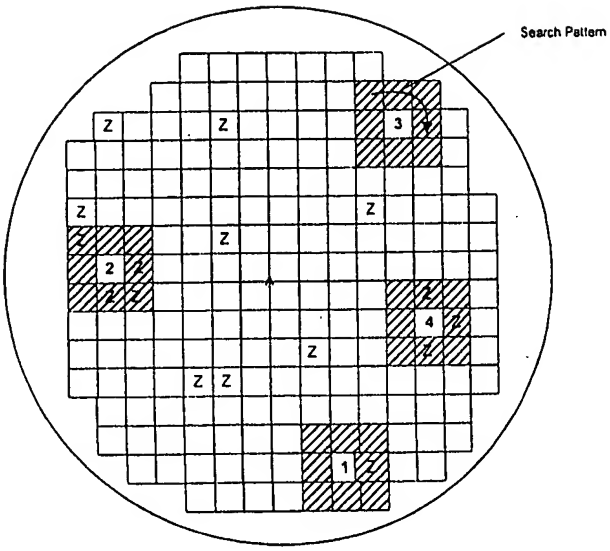
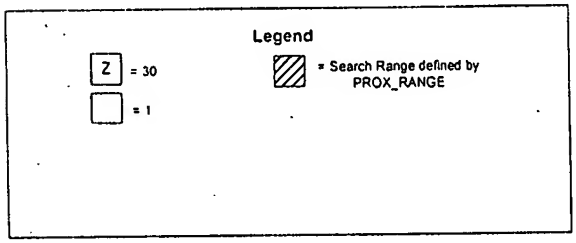

A
1726
COUNT > THRESHOLD ?
ASSIGN INK VALVE
1728
B
LAST DATA PRINT ?
END

FIG. 17B

Proximity Weighting Graphical





Description

Proximity Weighting reviews all data points in a data set arranged in x,y fashion and analyzes the points that have values within a specified range. Analysis consists of searching an area, radius of specifed units, around each device and looking for devices that are above a certain threshold. If a device is found that has a value above a said threshold, then it either lends a specified wieght to the device being analyzed; weight depends if the device is adjacent or diagonal to the device being analyzed.

Logic

Loop through each x,y location
If x,y location has value between UPPER_BOUND and LOWER_BOUND
Check all devices in the specified search range (see legend)
If device in search range being checked has a value that meets or exceeds the INFLUENCE_VALUE then
If device is adjacent, increment x,y location count by ADJACENT_VALUE else increment x,y location count by CORNER_VALUE
If x,y location is on edge of dataset then multiply by x,y location count by EDGE_FACTOR
if x,y location count exceeds THRESHOLD, assign x,y location a value of INK_VALUE

Examples

1. Count = 2, end result = 1
2. Count = 6, end result = 20
3. Count = 0, end result = 1
4. Count = 6, end result = 20

Parameters

THRESHOLD = 4 ADJACENT_VALUE = 2
CORNER_VALUE = 1 EDGE_FACTOR = 1
PROX_RANGE = 1 UPPER_BOUND = 30
LOWER_BOUND = 0 INK_VALUE = 20
INFLUENCE_VALUE = 30

FIG. 18

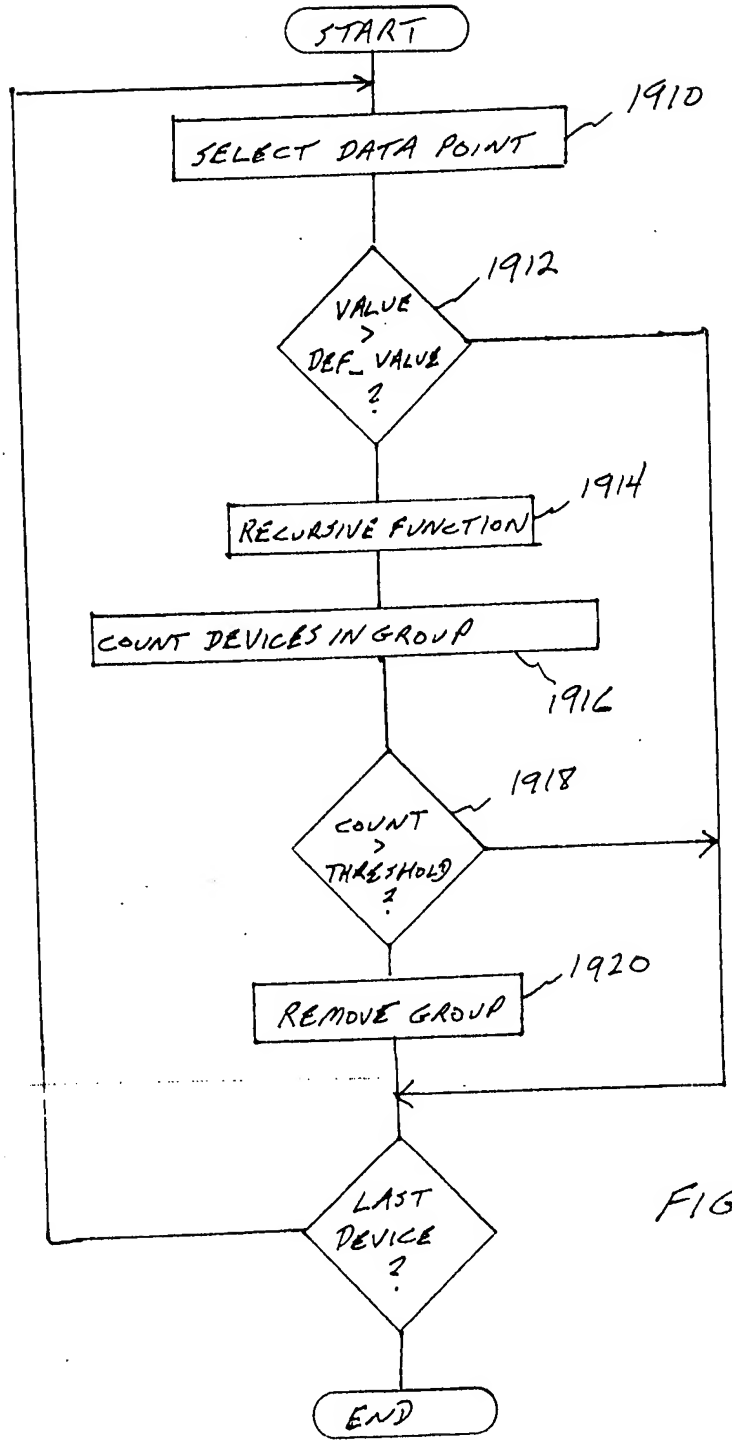
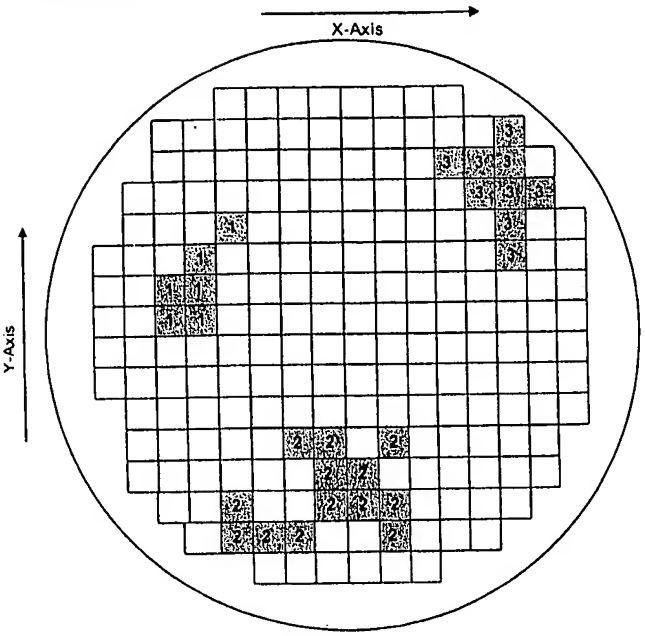

START
SELECT DATA POINT
1910
VALUE > DEF_VALUE ?
1912
RECURSIVE FUNCTION
1914
COUNT DEVICES IN GROUP
1916
COUNT > THRESHOLD ?
1918
REMOVE GROUP
1920
LAST DEVICE ?
END

FIG. 19

Cluster Detection



Description

The cluster detection function's purpose is to remove goups of locations (defined by the DEFINITION_VALUE) in datasets by assertaining how large each group is and if smaller than a said threshold, remove the group(s).

Logic

Loop through each x,y location
If there is a device at the x,y location
If x,y location has value equal to DEFINITION_VALUE
Call recursive function find next device in group and assign group index

Once group indexes have been assigned, count how many devices are assigned to each group
If group count is less than SIZE_THRESHOLD, remove it / set value to bin one

Examples

Group 1: Would be removed
Group 2: Would be kept
Group 3: Would be kept

Parameters

SIZE_THRESHOLD = 8
DEFINITION_VALUE= 30
INCLUDE_CORNERS = 1

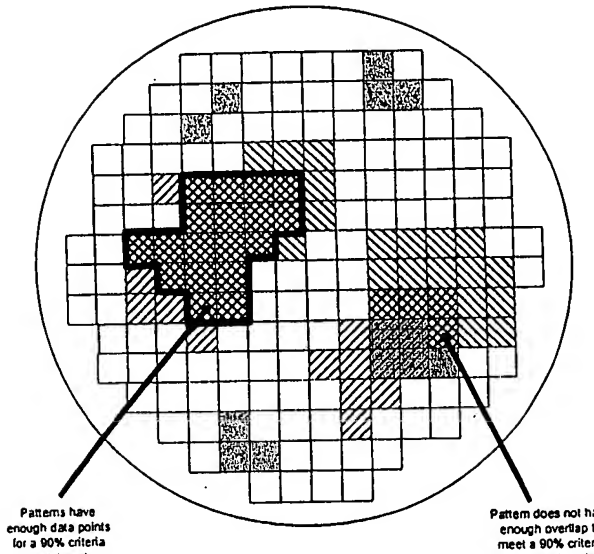
Legend

= Devices that have values matching the DEFINITION_VALUE

n = Group Index

FIG. 20

Percnelage of Pattern Composite Map Definition



Legend

- Tester Bin Failures in original data
- Failure Patter in Composite Map 1
- Failure Patter in Composite Map 2
- Composite Maps Overlap

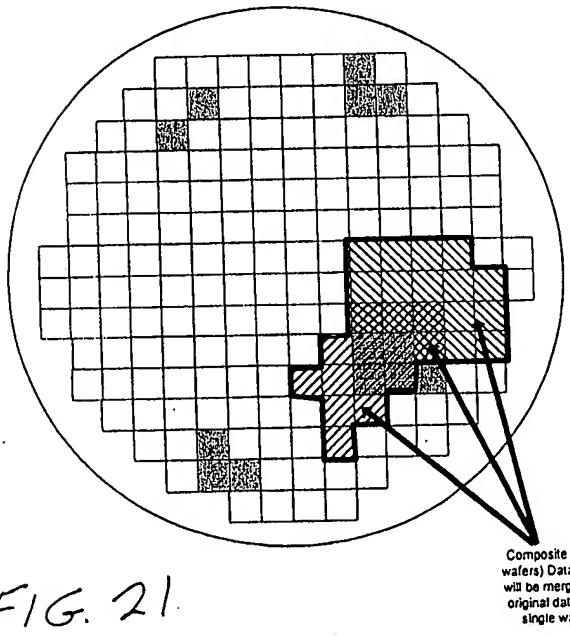
Description:

Looking at the example to the left, notice that there are two composite map patters, labeled one and two. Notice that there is an area where the two patterns overlap, it is this area and only this area of overlap that will be merged into the original data if and only if a user defined criteria is met. This criteria could be a fixed threshold, 15 for instance or a calculated value such as 90% of the data points constituting the patterns must overlap. In either case, the overlap on the upper left portion of the diagram would have sufficent overlap and would be merged but the overlap on the right of the diagram has only four devices and would not be merged.

Note: It is also possible that instead of merging only the overlapping patterns that all patterns are merged regardless of the criteria and the data points that do meet the criteria are only noted in an output file.

FIG. 24

Absolute Composite Merge



Legend

- Tester Bin Failures in original data
- Failure Patter in Composite Map 1
- Failure Patter in Composite Map 2
- Composite Maps Overlap

Description:

Looking at the example to the left, notice that there are two composite map patters, labeled one and two., any points from either pattern are merged in this type of merge operation.

FIG. 21

Composite Pattern Constitution

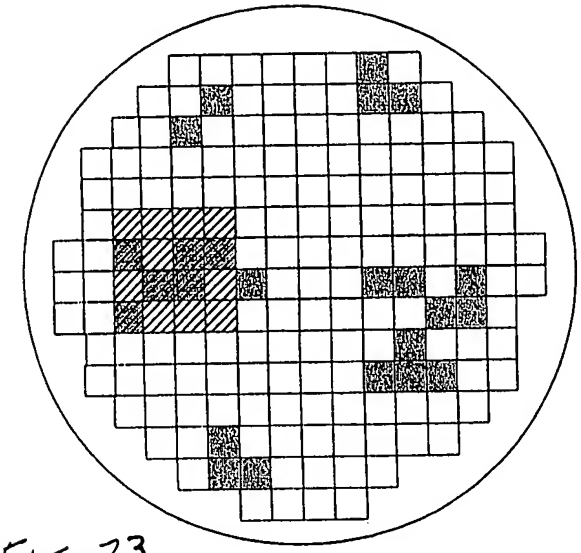


FIG. 23

Legend

- Tester Bin Failures in original data
- Failure Pattern in Composite Map

Pattern Constitution Mode

Given the example on the top left and assuming that the user has set the criteria for a pattern merge is that at least 50% of the composite map pattern must match the original data we can see that the composite pattern would not be merged on the originall data. This is due to the failure of the data to pass the criteria specified, at least 8 out of the 16 data point pattern must have corresponding point on the original data, of which there are only six. Since the pattern failed the criteria it will not be merged with the original data.

Abosolute Mode

In this mode, the example to the left would have the composite map super imposed over the original data map regardless of how much of the pattern is representitive of the actual data.

Multiple Recipe / Overlapping Composite Map Definition

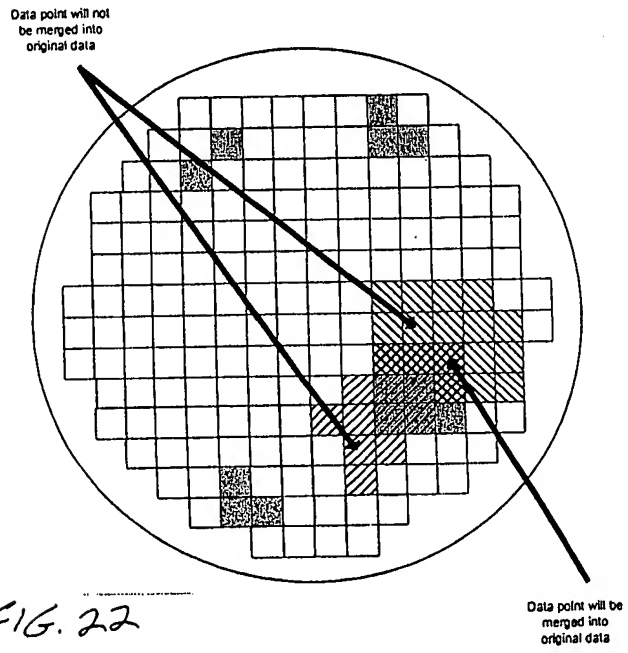


FIG. 22

Legend

- Tester Bin Failures in original data
- Failure Patter In Composite Map 1
- Failure Patter in Composite Map 2
- Composite Maps Overlap

Description:

Looking at the example to the left, notice that there are two composite map patters, labeled one and two. Notice that there is an area where the two patterns overlap, It is this area and only this area of overlap that will be merged into the original data, the rest of the pattern is descarded.

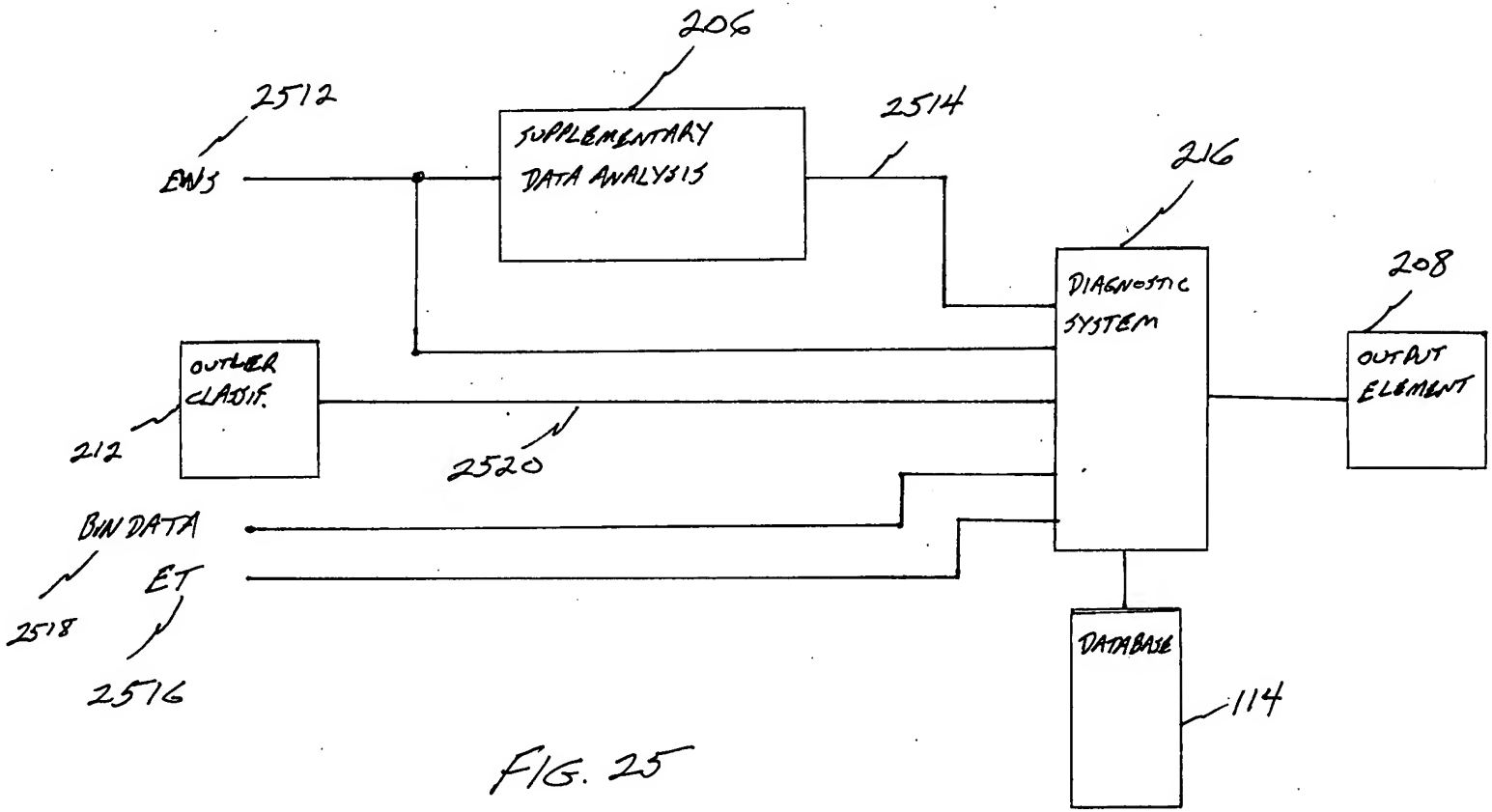

206
2512
EWS
SUPPLEMENTARY DATA ANALYSIS
2514
216
DIAGNOSTIC SYSTEM
208
OUTPUT ELEMENT
OUTLIER CLASSIF.
212
2520
BIN DATA
2518
ET
2516
DATABASE
114

FIG. 25

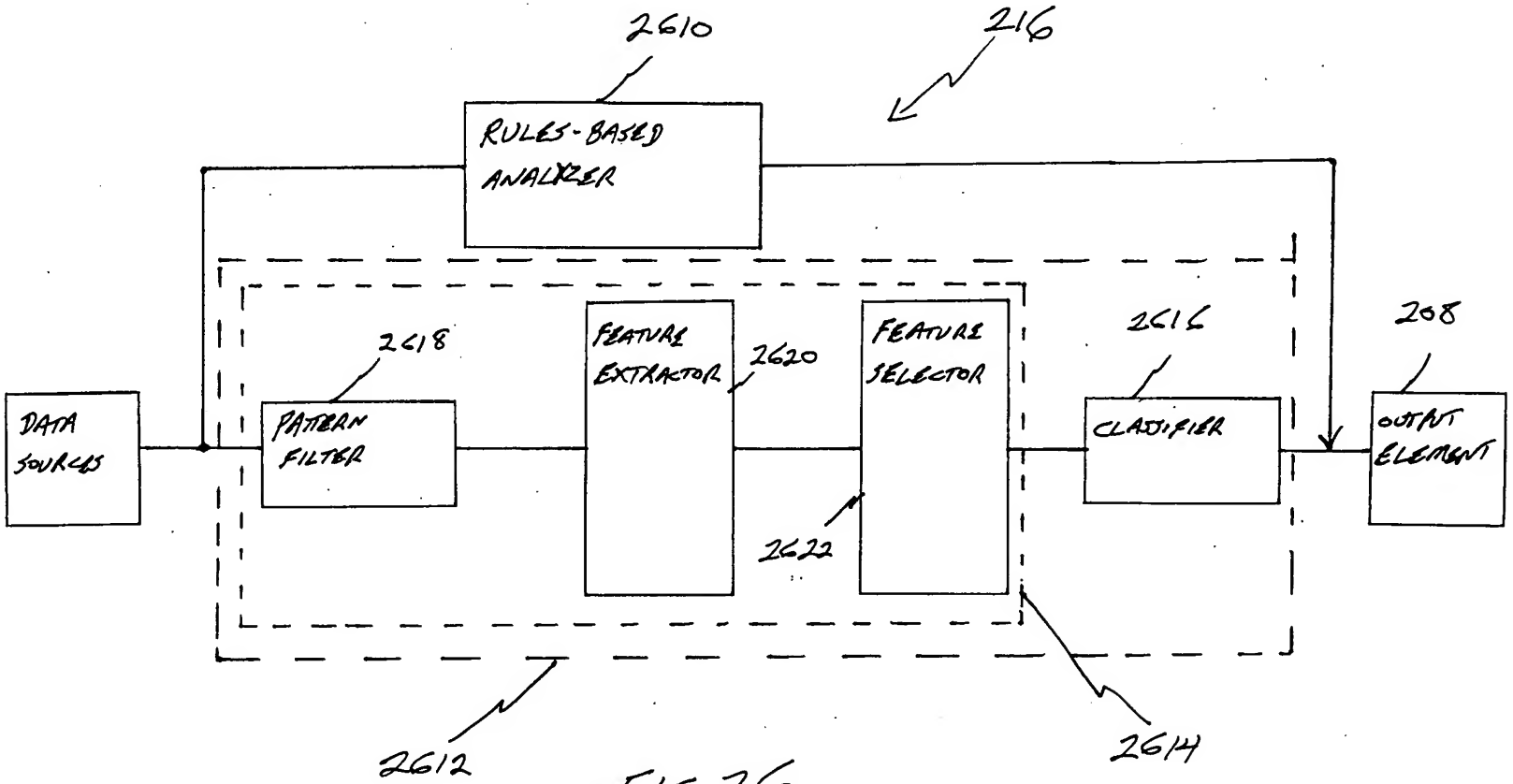

2610
216
RULES-BASED ANALYZER
2618
FEATURE EXTRACTOR
2620
FEATURE SELECTOR
2616
208
DATA SOURCES
PATTERN FILTER
CLASSIFIER
OUTPUT ELEMENT
2622
2612
2614

FIG. 26

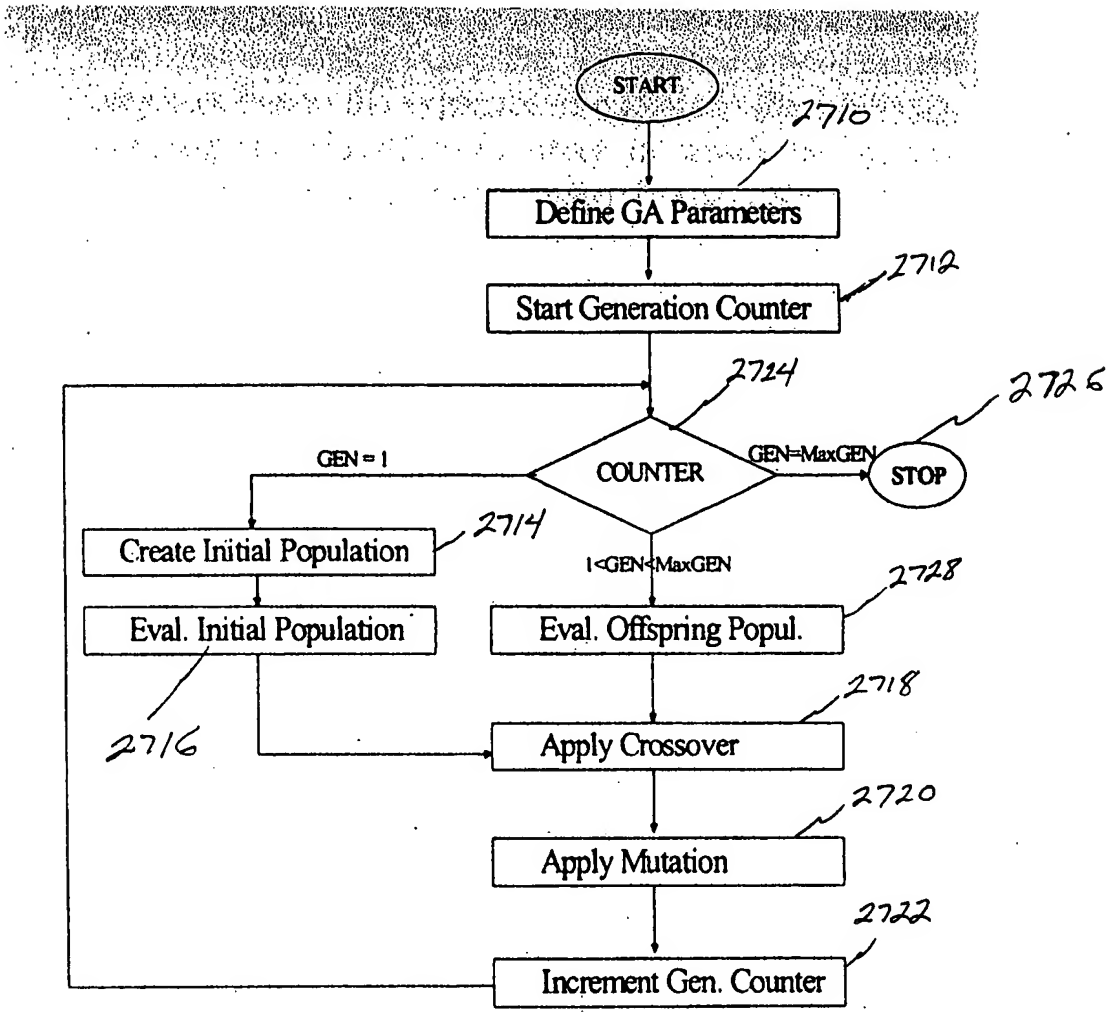

START
2710
Define GA Parameters
2712
Start Generation Counter
2724
2726
COUNTER
GEN = 1
GEN=MaxGEN
STOP
2714
Create Initial Population
1<GEN<MaxGEN
2728
Eval. Initial Population
Eval. Offspring Popul.
2716
2718
Apply Crossover
2720
Apply Mutation
2722
Increment Gen. Counter

FIG. 27

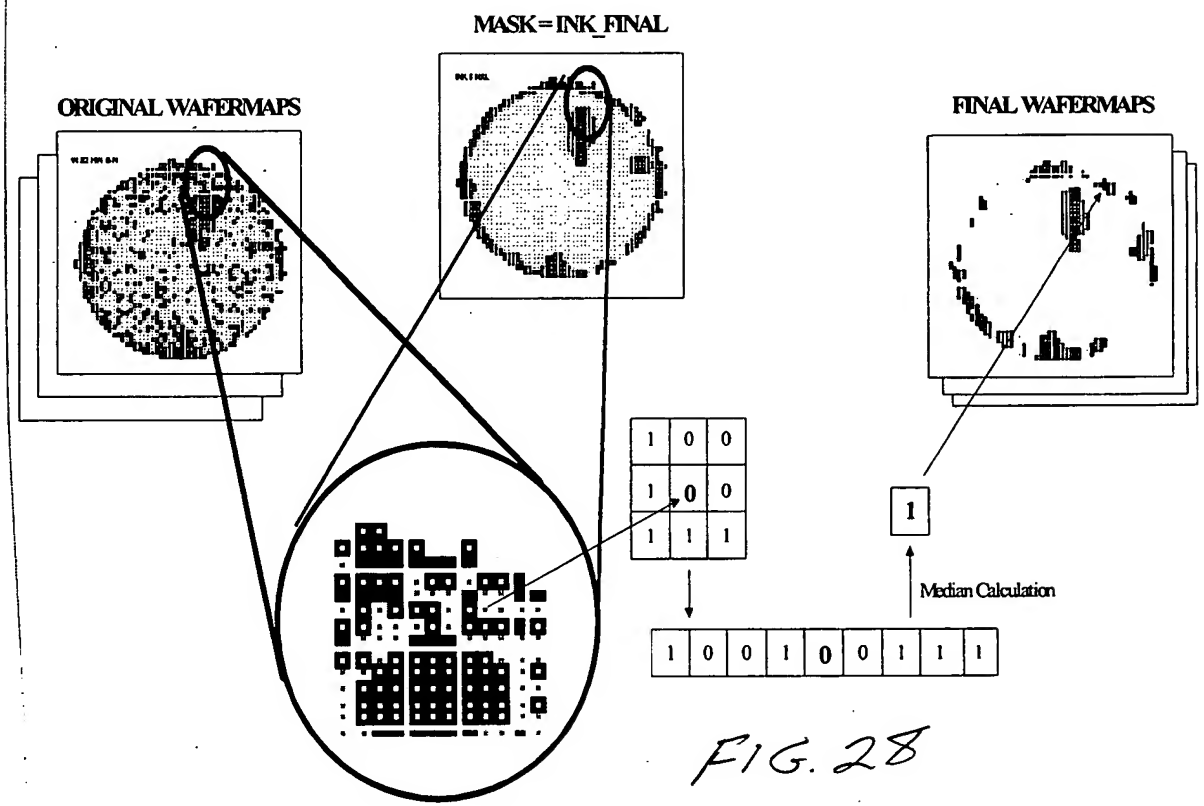

MASK=INK_FINAL
ORIGINAL WAFERMAPS
FINAL WAFERMAPS
1 0 0
1 0 0
1 1 1
1
Median Calculation
1 0 0 1 0 0 1 1 1

FIG. 28

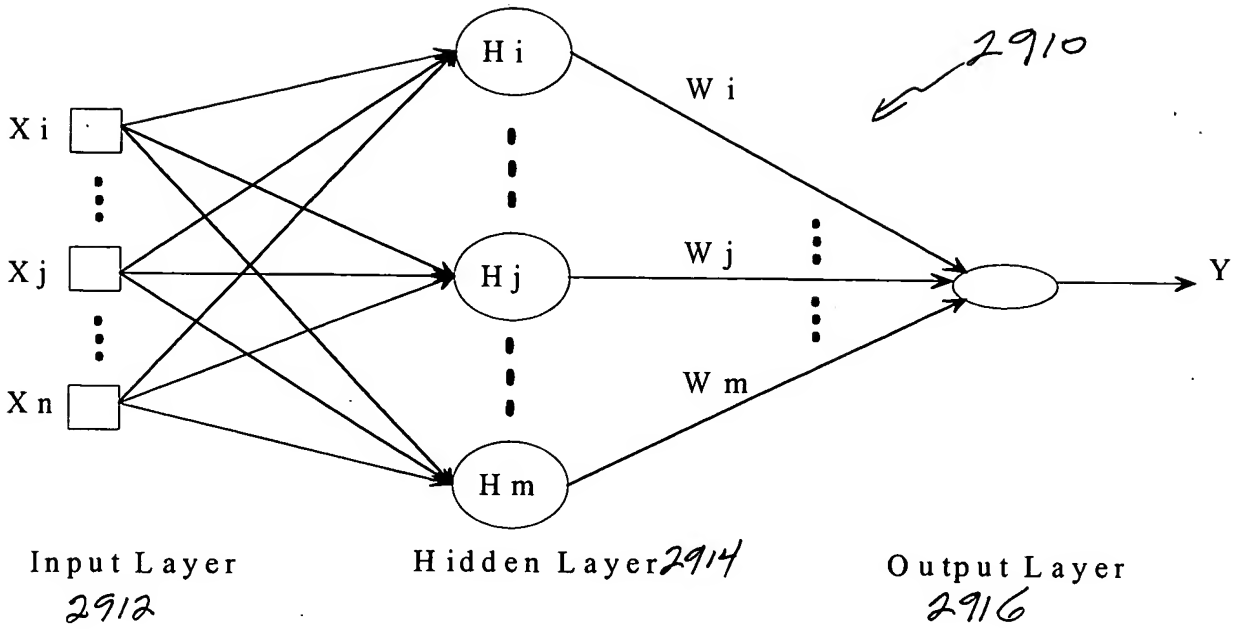

2910
Hi
Wi
Xi
Xj
Hj
Wj
Y
Xn
Wm
Hm
Input Layer
2912
Hidden Layer 2914
Output Layer
2916

FIG. 29